

レーダ観測データを用いた航跡管理のためのデータベース構築

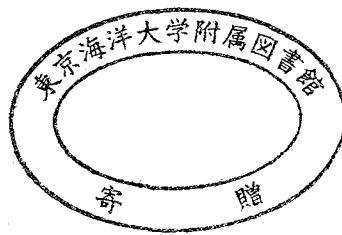
著者	中嶋 鮎子
学位授与機関	東京海洋大学
学位授与年度	2006
URL	http://id.nii.ac.jp/1342/00000825/

修士学位論文

レーダ観測データを用いた航跡管理のための
データベース構築

平成 18 年度

(2007 年 3 月)



東京海洋大学大学院

海洋科学技術研究科

海運ロジスティクス専攻

中嶋 鮎子

目次

1. はじめに	4
2. データベースの概要	5
2.1. 海上交通観測のニーズ	5
2.2. データベース構築の概要	7
3. 航跡抽出	8
3.1. リモートレーダネットワークシステム	8
3.2. レーダ観測データの性質について	12
3.3. 背景差分法を用いたグラウンドクラッタの低減	14
3.3.1. 背景差分法	15
3.3.2. 動的な背景差分法	16
3.4. 航跡抽出	18
3.4.1. 2値画像処理	19
3.4.2. 膨張・収縮	21
3.4.3. ラベリング	24
3.4.4. 対応付け	26
4. データベースの実装	28
4.1. 開発環境	28
4.2. GUIによる実装	28
4.3. 海上交通観測のための航跡検索	30
5. 木構造を用いた航跡データの管理	32
5.1. 木構造の航跡データへの適用	32
5.2. 代表的な木構造	34
5.2.1. B-Tree	35
5.2.2. R-tree	36
5.3. 領域分割と木構造	37
5.4. R*-Treeによるインデックスの作成	39
5.4.1. 木構築	39
5.4.2. R*-Treeの特徴	42
5.5. 実験 線形探索との時間的効率における比較	43
6. おわりに	52

図目次

Fig.1	データベースの概要図	7
Fig.2	東京湾リモートレーダネットワークシステム	8
Fig.3	レーダ映像（レーダ観測データ）伝送システムの構成図	9
Fig.4 (a)	防衛大学校に設置されたレーダアンテナ	10
Fig.4 (b)	防衛大学校屋上のレーダ局の収納ボックス（レーダ局の通信制御装置）	11
Fig.5	監視局の通信制御装置	11
Fig.6	レーダ観測データ（B-scope 形式の画像・6 マイルレンジ）	13
Fig.7	背景差分処理結果	15
Fig.8	レーダ観測データ（ m ）	17
Fig.9	動的な背景（ m_{μ} ）	17
Fig.10	航跡抽出のフローチャート	18
Fig.11	2 値画像処理の一般的な流れ	19
Fig.12	膨張・収縮処理の概念図	21
Fig.13	動的な背景差分処理、2 値化処理、膨張収縮を行ったレーダ観測データ	23
Fig.14	ラベリングの例	24
Fig.15	最近傍法による船舶のフレーム間対応付け	26
Fig.16	航跡の抽出結果	27
Fig.17	データベースの基本操作画面	29
Fig.18	矩形内を通過する船舶の検索結果	30
Fig.19	通過した船舶の隻数表示	30
Fig.20	航跡のプロパティ表示	31
Fig.21	航跡データのデータ構造イメージ図	32
Fig.22	木構造の概念図	34
Fig.23	R-Tree と MBR	36
Fig.24	航跡データ分割の概要図	37
Fig.25	航跡データの木構造への格納	38
Fig.26	矩形領域の最小化	39
Fig.27	オーバーラップの最小化	40
Fig.28	オーバーフロー	41
Fig.29	実験に使用した航跡データの例	44
Fig.30	インデックスの大きさによる検索時間の推移	51

表目次

Table1	46 - 47
Table2	49

はじめに

東京湾は一日あたり約 1100 隻もの船舶が航行する世界有数の輻輳海域である。湾内は他の海域に比べて水深が浅く、港の数も多いため航路は複雑である。さらに航行する船舶の種類も多様で、内航船・外航船を問わず大型タンカー、コンテナ船、客船、漁船、プレジャーボートなどが行き交い、これらの船舶は各々の役割から海上での振る舞いも大きく異なる。よって航路内や湾岸は非常に過密な状況と言える。近年では海難事故も多数発生し、増加傾向にあることから、安全への対策が急がれている。

海上の安全策を検討する手段として、本研究では卒業論文に引き続き東京湾岸に設置したレーダから湾内の状況を観測し、画像処理手法を適用した船舶検出・追跡システムの研究を進めてきた。レーダから得られる時系列の信号であるレーダ観測データに画像処理を行い、航行中の船舶を移動物体として正確に検出・追尾することが目標である。得られた画像データは「航跡」として扱われ、いつ、どの海域に船舶が存在したか、あるいは船舶がどのようなコースを辿って出発地点から目的地点へ移動するかなどの多くの情報を含むため、海上交通における危険要因を分析し、安全策の考案に役立てられると考えてきた。

しかしながら、レーダ観測データからの航跡の自動抽出は、レーダの特性による障害や、パターン認識処理過程で生じる検出漏れ、誤検出、とぎれなどの現象が避けられず、完全な航跡を抽出することは困難であった。そこで本研究ではレーダ観測データから自動的に得られた航跡データをデータベースに入力し、そこから個々や群における航跡の動向をつかむことを目指した。このデータベースを用いることで、膨大な数に上るデータを管理し、また航跡データの中には不完全な追跡結果を含んでいるということを前提とした上で、そこから海上交通解析に有益なデータを取り出すためのツールとして使用できるよう試みた。主な機能としてはデータベースを用いて大量の航跡データ中から固有の海域や時間帯の航跡だけを効率よく抜き出したり、あるいは長時間、長期間にわたる航跡の傾向を提示することである。各航跡の詳細データや個別で表示するほか、ある海域における船舶数を調べ海域別交通量調査のためのデータとしたり、また海難事故などが発生した場合に直前の時刻で周辺船舶の動きを抜き出すなど、海上交通の特性を把握するためのツールとして使用することを目的とする。

またデータベースに入力する航跡データのデータ量は日ごとに千単位の割合で増加する。航跡の管理・検索は航跡データが増えるほど時間を要するため、検索に掛かる時間的コストを効率化する必要がある。本研究では時間と緯度経度からなる空間データである航跡データに木構造を用いたインデクシングを行い、管理と検索を行った。

本論文ではこのような機能を備え、GUI (Graphical User Interface) による操作を行い、かつ航跡の高速検索が可能なデータベースを提案する。

1. データベースの概要

1.1. 海上交通観測のニーズ

海上の安全性を向上させるため、これまでも海難防止対策が数多く研究・考案されてきた。現在では日々進歩している無線通信技術を用いた船舶の航行支援によって、より高度なシステムが開発されている。すでに実用化されているものとしては、レーダ信号を用いて付近の船舶の進路予測を行い、衝突の危険を警告する ARPA や、自船の航海情報を多船もしくは陸地に向けて発信し、安全に役立てる AIS などが代表的である。AIS については IMO (International Maritime Organization) によって大型船舶に限り搭載が義務化されている。さらに研究段階においても海難事故の原因が見張り不十分によるものが多い点から、カメラによる見張り支援システム⁽¹⁾の開発が進んでいたり、船舶自体を陸から遠隔操船し、交通流を管理する研究も行われている⁽²⁾。

一方で、海域における船舶交通の特性から海難の危険性を高める要因を解析し、把握する方法もある。船舶の交通解析にはレーダで観測したデータを人が目視で追跡して求めていたのが主流であったが、この方法は処理対象となるデータ量が非常に多く、またその数は日ごとに増大するため自動化が必須である⁽³⁾⁽⁴⁾。レーダ観測データにはターゲットである船舶以外の反射 (Clutter) が多く映っており、判別が難しいことに加えて、移動する船舶を追跡し続ける際にも途中で船を見失ってしまう「ロスト」や、異なる船舶を同一として追跡してしまう「乗り移り」が多く発生し、航跡の自動抽出には様々な試行錯誤がなされてきた⁽⁵⁾。人手に頼らず自動的に船舶の航行状況を観測し、蓄積する手段の確立が可能になれば、海上での船舶の動きを記録・蓄積したデータを解析することによって、海上交通システムの向上や港湾設計、航路設計の見直しに役立てることが大いに考えられるためである。本研究でも海上の安全策のひとつとして、このように蓄積データを用いて海上交通の動向を正確かつ継続的に把握・解析できるシステムの開発を目指す。

海上交通観測の基本は正確な航跡を得ることである。この航跡追尾の目視作業には莫大な作業を要し、自動化が強く望まれる。しかし前述したとおり、レーダ観測データから完全な航跡を取得することは、現在の技術ではレーダ信号から完全な航跡を得ることは非常に困難であり、画像処理・認識の問題として考えると船舶の検出・追跡の精度を向上させることはできるが、100%の認識率はほぼ不可能に近いと言える。また、仮に統計的に99%の抽出が得られることが分かったとしても残りの1%の誤りは目視による誤り訂正を行わない限り、正しく抽出された結果と区別がつかない。

しかし、海上交通に関わるニーズによっては、航跡が完全であることは必ずしも必要とは限らない。そのようなニーズの例として、海上交通の動向把握、交通量調査などが上げられる。仮にひとつの航跡に途切れが生じていたとしても、ある地点での船舶の通過隻数を求めることは可能であり、また海域による船舶の密度や航跡の特徴などは個々の航跡の正確さには囚われにくい。これらのデータを自動化して大量に得ることにより、航路設計や港湾設計の見直しに必要なデータとして蓄積することができる。

この研究では様々な時間帯の航跡を自動的に収集し、同時にデータベース化することによって、解析に必要な航跡データを検索・表示し、このようなニーズに応えようとするものである。

1.2. データベース構築の概要

1 章で前述したとおり、本研究で提案するデータベースは、レーダ観測データから自動的に抽出した航跡データを用いている。レーダ観測データは東京湾岸に設置されたレーダから遠隔操作で取得したもので、湾内を一定時間間隔でセンシングした時系列データである。使用したデータに関しては 3 章で述べるが、本論文ではこのデータを擬似的に動画像と見なし、画像処理手法を適用した処理で移動する船影を追跡し、航跡データを得ている。さらにこれをデータベースに投入するが、ここで取得した航跡データは膨大な数であるため、航跡を円滑に検索できるよう木構造を用いてインデックスを作成し、管理している。これについては 5 章で後述する。提案するデータベースは海上交通解析のためのツールであることを目的としており、任意の範囲（緯度・経度）と時間で航跡の検索をした後、それをすばやく表示することに主眼を置いている。上記のように木構造を用いた航跡管理を行うほか、4 章で記述する GUI を用いて実装を行い、各航跡の表示や操作が簡易に行えることを目指した。レーダによって得られた情報は最終的にデータベース上から操作し、これを用いて必要な航跡データを検索・表示することで海上交通解析のためのデータを取得することを目指している。Fig.1 にレーダ観測データから航跡データがデータベースに投入されるまでの概要図を示す。Fig.1 では図中左上に記載されている東京湾南部の地図が示す灰色の海域をセンシングした場合を示しており、レーダ信号データが航跡データとして加工され、インデクシング後にデータベースへ投入される様子を示している。

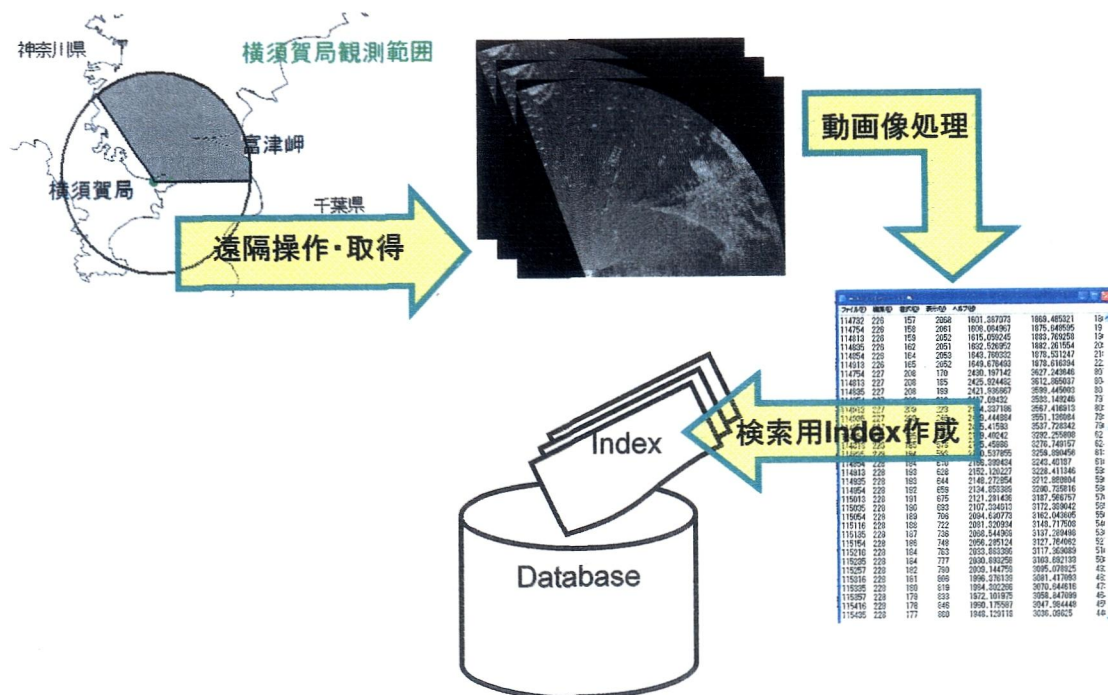


Fig.1 データベースの概要図

2. 航跡抽出

2.1. リモートレーダネットワークシステム

本研究で扱うデータは、前述の通り東京湾岸に設置されたパルスレーダによって取得される。レーダは神奈川県川崎市の東扇島（以下扇島データ局）と神奈川県横須賀市の防衛大学（横須賀レーダ局）の二箇所に設置されているが、現段階では横須賀レーダ局からのみ取得している。レーダ観測データなどのレーダから得られる情報は、ネットワークを介して東京海洋大学海事交通共同研究センターの監視局(以下、東京海洋大学と記す)に伝送され、蓄積される。レーダの電源管理、レーダの照射する角度や時間間隔などのデータ取得のためのパラメータは、ネットワークを通じてすべて東京海洋大学構内でコントロールされている。Fig.2 にリモートレーダネットワークシステムの概念図、Fig.3 にレーダ観測データを伝送する際のネットワーク構成図、Fig.4 (a), (b)にレーダ局の様子、Fig.5 に監視局の様子を示す。

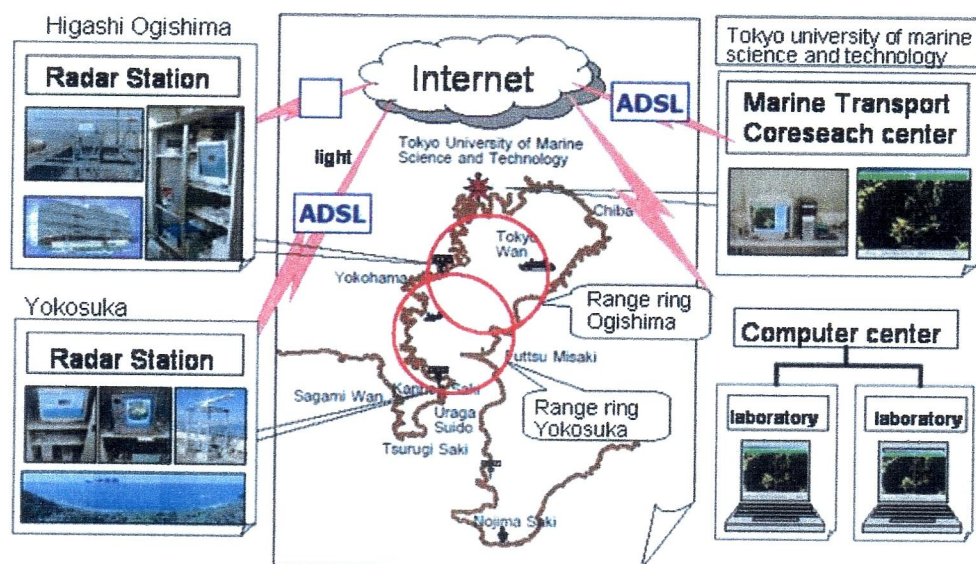


Fig.2 東京湾リモートレーダネットワークシステム

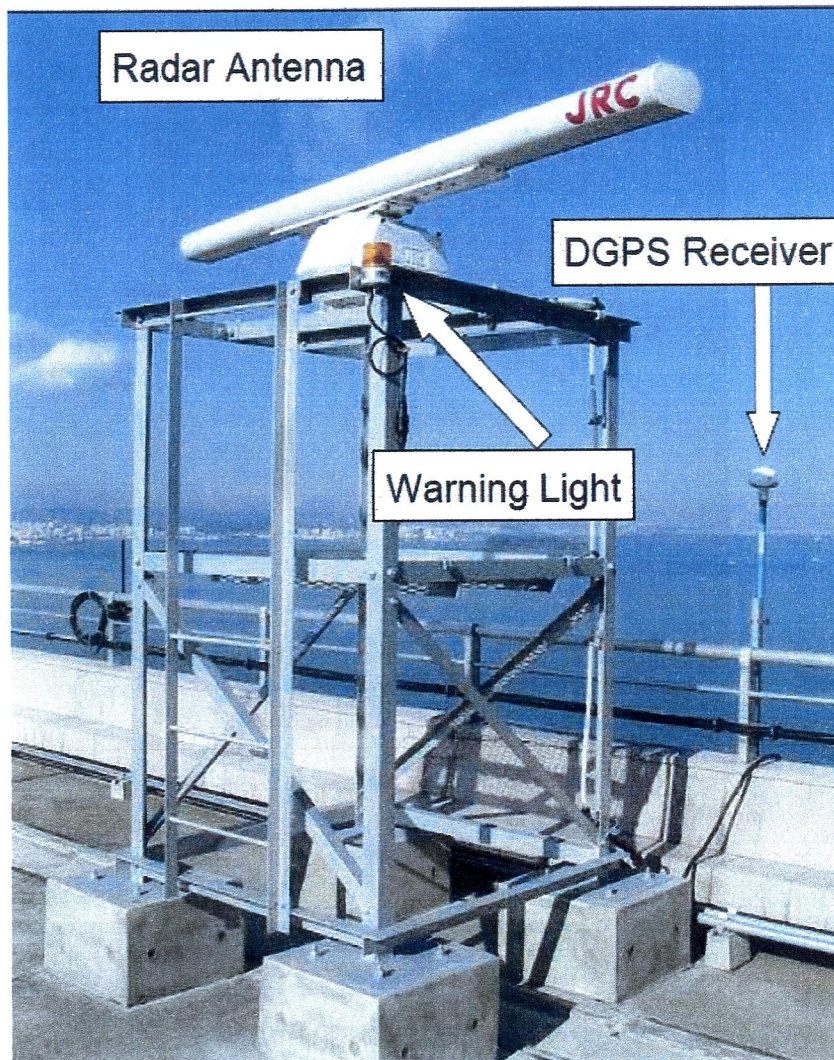


Fig.4 (a) 防衛大学に設置されたレーダアンテナ

Fig.4 に示したレーダの仕様は以下の通りである。

- JRC 社製 パルスレーダ 高さ約 519mm×回転直径約 2850mm
- 指向特性
 - 水平ビーム幅 0.8° 垂直ビーム幅 25° 以下
 - サイドローブレベル $\pm 10^\circ$ 未満 -26dB 以下 $\pm 10^\circ$ 以上 -30dB 以下
- 送信パルス幅（本研究に用いている値） $0.8 \mu s / 781Hz$
- 回転数 約 20rpm
- 送信出力 $25kW \pm 50\%$
- 送信周波数 9GHz 帯
- ARPA 機能付
- DGPS 受信機付

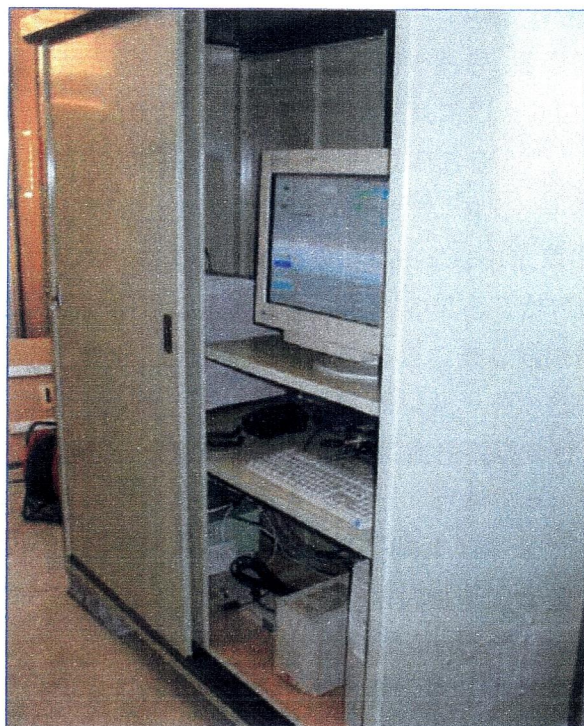


Fig.4 (b) 防衛大学校屋上のレーダ局の収納ボックス
(レーダ局の通信制御装置)

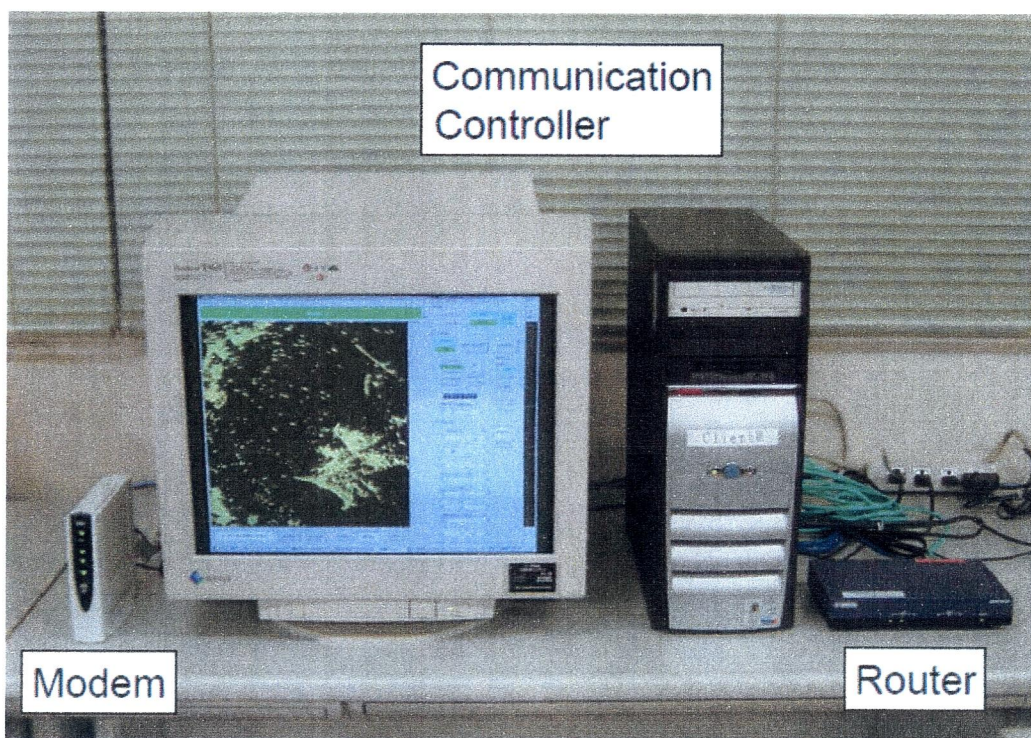


Fig.5 監視局の通信制御装置

2.2. レーダ観測データの性質について

センシングする範囲は方位 330 度から 090 度の間である。横須賀レーダ局から観測した場合、浦賀水道航路と中ノ瀬航路の分岐点、また東京湾の入り口に当たる地点を含み、東京湾を出入りするすべての船舶が把握できる海域である。

Fig.6 に示すように、今回用いたレーダ観測データは、B-scope といわれる表示形式の画像データで、ビットマップ形式の画像である。レーダは方位角を変えながらパルスを当てることにより物体の位置情報を把握していくものであるが、B-scope 画像はその放射状のデータを横に並べ、パルスの反射強度を明るさで表し画像化したものである。したがって横軸は時間ないしはレーダの方位角、縦軸はレーダからの距離（レンジ）となる。通常の緯度経度に対応させるためには処理後に座標変換を行う必要がある。

パルスの送信および受信は空中線が回転する過程で行われる。この空中線が観測範囲内を一回転した分をスイープと呼ぶが、B-scope 画像は 1 スイープを一枚の画像に表現したものであるともいえる。1 スイープあたりの時間はおよそ 2~3 秒である。従って後に示す画像形式のレーダ観測データはカメラ画像などと違い x 方向の位置によりセンシングされた時刻は異なるが、観測範囲に対する船舶の動きは数秒単位ではほぼ変わらないと考え、本研究ではこれを一枚の擬似画像と見なす。

本研究で用いるレーダ観測データのレンジは 6mile~16mile までのデータを用いた。レーダ観測データでは座標変換を行った場合、レーダからの距離が遠いほど横方向（同一距離上にある異なる角度の地点）の距離が離れてしまう。従って画像データは平面的に均一な情報量を持つわけではなく、レーダに近い側が密であり、離れるほど疎であるという性質がある。よって船舶抽出を行う場合にはレンジを必要以上に大きくすることは望ましくない。本研究では将来的に、複数のレーダ同士でセンシングした海域が重なった部分で情報を補うという点から、今後は扇島レーダ局と横須賀レーダ局のセンシング海域が重なる 9mile レンジのデータを中心に扱う予定である。

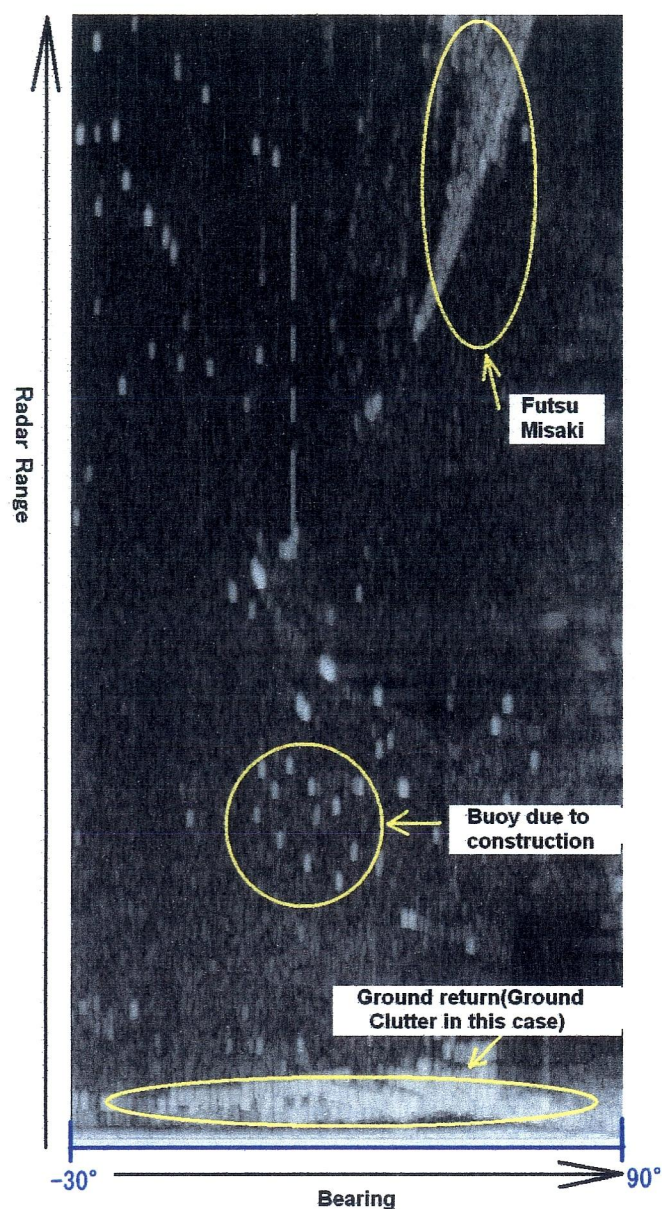


Fig.6 レーダ観測データ (B-scope 形式の画像・6mile レンジ)

一般的にレーダのパルスを反射しやすい船舶などの金属物質、また安定して反射する陸地部分などは反射強度が高い。本研究で用いるレーダ観測データは、反射強度を 256 段階の輝度値で表し、強いほど白く表現している。Fig.6 において黄色円で括った画像内下部の白い部分はレーダの設置場所付近の陸地、右上の三角形状の部分は千葉県富津岬である（これらは Grand Clutter と呼ばれる）。さらに白い点状の部分のうちいくつかは船舶の映像、もしくは海上に設置されたブイなども含まれている。また輝度値（明るさ）の低い霧状の反射は主に Sea Clutter と呼ばれる海面からの反射、もしくは Weather Clutter と呼ばれる雨雪や空中浮遊物等の反射によることが多い。

2.3. 背景差分法を用いた Grand Clutter, Sea Clutter の低減

本研究においてさきに開発していたビットマップ形式のレーダ観測データによる航跡自動抽出は、動画像における特徴的な点を移動ベクトルの分布として表したオプティカル・フローの推定を基本にしていた⁽⁶⁾。しかしこの方法は、本来ビデオ画像を対象としたものであり、レーダ画像特有の海面や大気浮遊物による反射（Sea clutter, Weather Clutter, レーダビーコンの反射など）や陸地などの静止物体からの強い反射（Grand Clutter, ブイの反射など）が障害となり、適用するデータによっては十分な航跡抽出ができなかった。そこで本論文に使用したデータベースに用いる航跡には別の方法⁽⁷⁾を試みた。

本論文では処理対象が蓄積されたデータであるという観点から Pless らが提案した動的な背景差分法による処理⁽⁸⁾を行った。本章では 3.3.1 で背景差分法について、3.3.2 で画像に含まれるランダムのある Clutter への耐性に優れた動的背景差分による Grand Clutter・Sea Clutter の低減と、実際のレーダ画像への適応結果について述べる。

2.3.1. 背景差分法

画像処理分野における移動物体の検出法の一つとして代表的なものに、背景差分法がある。背景差分法あらかじめ背景となる画像を取得しておき、入力画像との差分を求めることにより、背景から大きく変化した領域を処理対象領域として抽出する手法である。背景が既知の場合、差分画像 $f(m)$ は次式によって生成される。

$$f(m) = (m - m_{background})^2 \cdots \cdots [1]$$

この場合画像処理では一般的にあらかじめ用意された背景に侵入した物体の検出に用いられることが多い。例として Fig.7 に背景差分の処理画像を示す。



Fig.7 背景差分処理結果

レーダ観測データは取得する画像ごとに船の位置が変化し、同じ物体をセンシングしても天候条件などにより映り方が異なる。背景差分法は背景が固定の場合画像内の全体的な変化に弱いため、「背景」は既知のデータではなく、複数のレーダ観測データから推定した背景画像を動的に定義する必要がある。

2.3.2. 動的な背景差分法

Pless らは背景の作成法を画像の条件に合わせ複数提案しているが、本研究ではレーダ観測データの中では比較的安定して反射(輝度値)が見られる Grand Clutter に注目し、以下の手法を用いた。

$$\left. \begin{aligned} m &= I \\ f(m) &= |m - m_{\mu}|^2 \\ m_{\mu_{\text{new}}} &= \frac{n-1}{n} m_{\mu} + \frac{1}{n} m \end{aligned} \right\} \dots\dots\dots [2]$$

ここで I はレーダ信号 (レーダ観測データ)、取得したレーダ観測データが n 枚目の時とする。 μ は画像データの平均を表し、平均された画像 $m_{\mu_{\text{new}}}$ は逐次生成される動的な背景とする。レーダから得られる m に対し逐次 $f(m)$ を計算し、これを次節の追跡処理の入力画像とする。

Fig.8 に m であるレーダ観測データを示す。ここには船舶や陸岸反射(Grand clutter)のほかに、航跡検出にとって障害となる海面反射(Sea Clutter)、雨雪などの浮遊物の反射(Weather clutter)が含まれるが、式[1]による動的な背景生成を行えば、時間的に信号値のばらつきが大きければ Fig.9 に示したように低減できる。

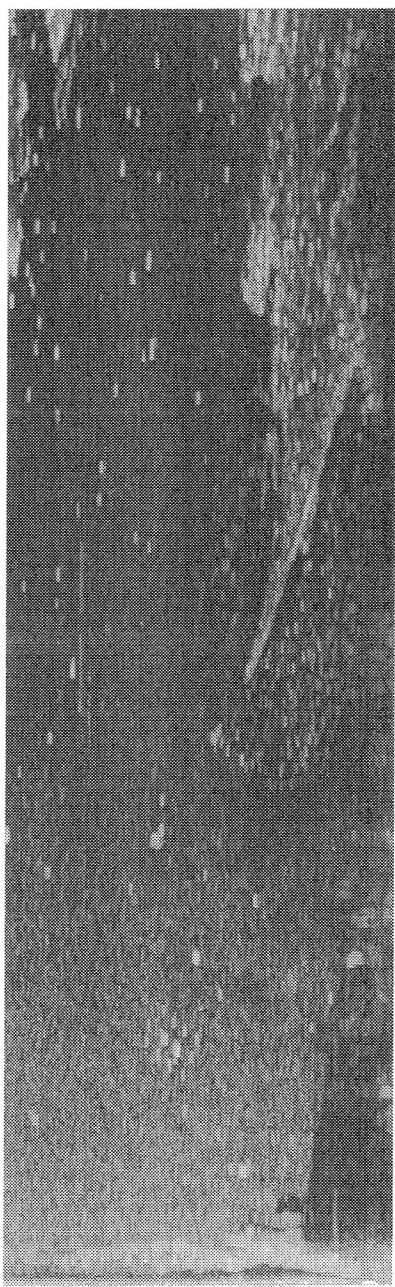


Fig.8 レーダ観測データ (m)



Fig.9 動的な背景 (m_d)

Fig.8, Fig.9 は共に横軸は方位角もしくは時間で方位 330 度から 090 度の間、縦軸はレーダからの距離で 9 マイルとなっている。Fig.8 と Fig.9 の差分画像は航行する船舶の候補領域であり、前述したとおり 3.2 で述べる追跡処理の入力画像となる。

2.4. 航跡抽出

前節で記述した方法によって生成した動的背景を用い、その差分画像を本節で述べる船舶抽出のための入力画像とする。

航跡抽出のためのアルゴリズムは差分画像の2値化、モフォロジー演算（膨張・収縮処理）、ラベル化により得たラベル化画像列に対し、有意な画素数の領域のみを追跡対象として検出している。対応する領域は、前の時刻（フレーム）における最近傍領域とした。Fig.10 に航跡抽出のフローチャートを示す。Fig.10 では灰色の処理が本章で述べる航跡抽出アルゴリズムである。

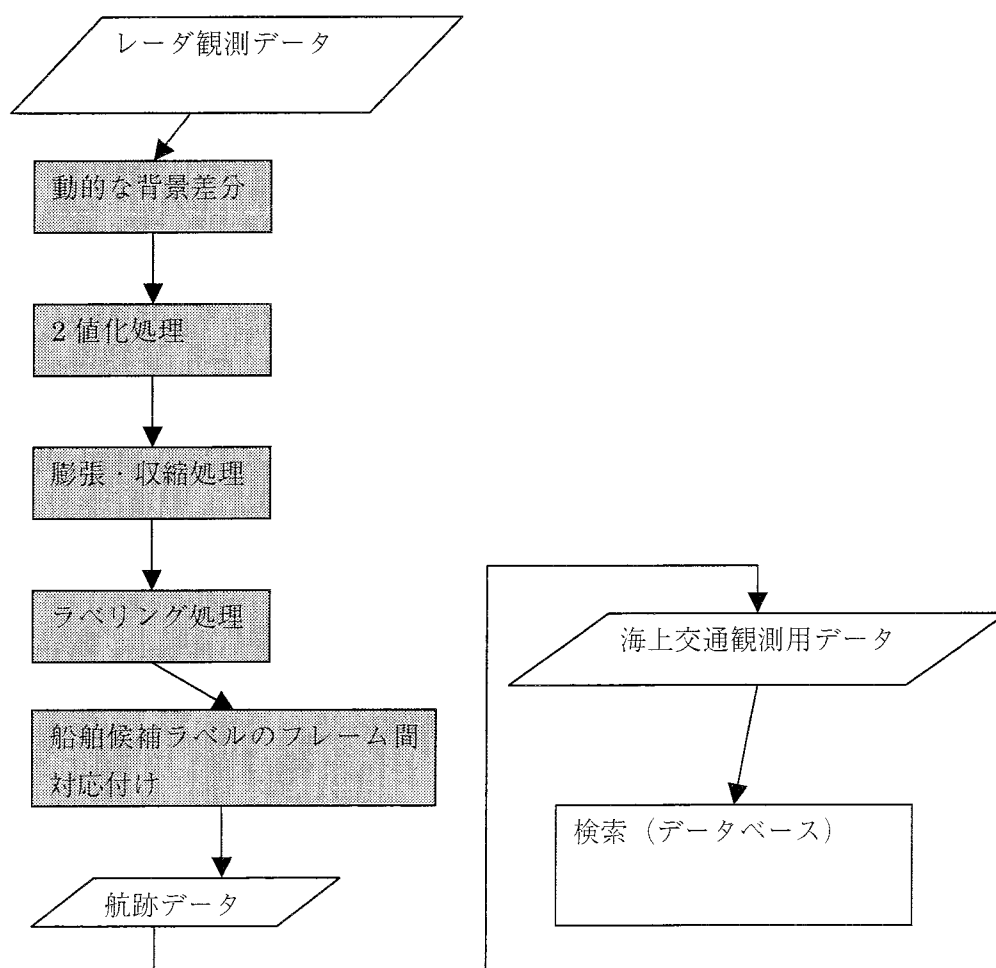


Fig.10 航跡抽出のフローチャート

2.4.1. 2 値化処理

濃淡が無く、白と黒しかない図形・画像は、画素の値が 0, 1 の 2 つの値しか取らないので、2 値画像と呼ばれる。2 値画像処理は、コンピュータ画像処理の中でも特異な位置を占めている。最も体系だって研究された分野であり、実用的な画像処理システムで最も熱心に用いられてきた技術でもある。これは主として以下の理由によるものである。

第一に、文字や図面の画像など、画像処理の対象には背景と図形からなる観念的に 2 値の画像が多く存在する。このような画像は観測時（画像データ）で多値で入力されていたとしても、2 値化画像処理を適用するのが合理的である。第二に、濃淡画像やカラー画像の処理においても、2 値画像への変換と解析を経て対象物やシーンを解析・認識する場合が多い。第三に図形の形だけを扱う 2 値画像処理では、幾何学的な諸概念ができていたので、濃淡画像に比べて理論や手法の体系化が進んでいる。

Fig.11 に 2 値画像処理の一般的な流れを示す。本研究でも同様な処理を行うため、入力画像に 2 値化を採用している。

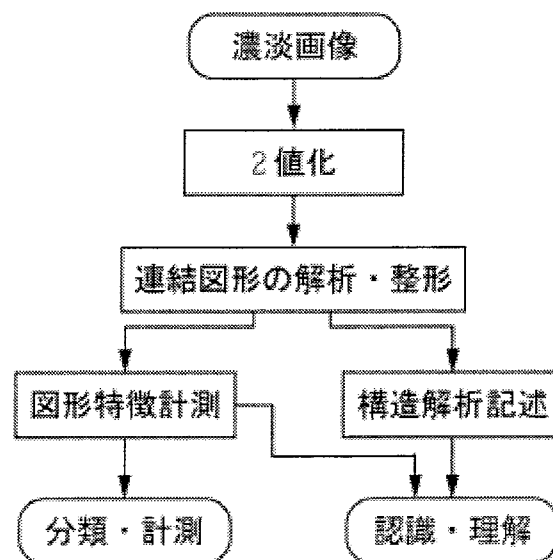


Fig.11 2 値画像処理の一般的な流れ

デジタル画像の場合、濃淡画像を 2 値化して図形の部分を値 1 を持つ画素、背景部分を値 0 を持つ画素として表した 2 値画像に変換する。この 2 値画像は一般に複数の連結図形の集合となるので、図形の特徴抽出や解析を容易にするために、連結図形の解析と

整形操作を施す。この研究では連結した画素の塊を 1 つの塊（主に船舶）として処理するためにラベリングという処理を行う。ラベリングについては 3.4.3 で後述する。

2 値化処理には p-タイル法やモード法、判別分析法など多くの方法が提案されているが、これらは 2 値化する際の閾値（濃淡画像の画素をどの値で白、黒とするかの輝度の判定基準値）を決定するための手法である。輝度値とは画素が持つ明るさの値のことを言う。本研究ではレーダ観測データ取得の際にできる限り条件を一定に保つため、レーダ信号のゲインを一定に観測した。背景差分処理を行った後のデータは **Sea Clutter** や船舶の反射が含まれているが、これらを完全に切り離すことは不可能であるので、主に船舶の反射を含む輝度値である 50 を閾値とし、2 値化処理を行った。

2.4.2. 膨張・収縮処理

3.4.1 では画像処理技術における 2 値画像の概念と、2 値画像を適用した後の一般的な処理の流れについて述べた。Fig.11 では 2 値化処理により得られた画素の塊はラベリングにより関連付けられるとあるが、実際の処理画像には誤差が生じるため 2 値化処理を行うと塊の中に穴ができたり、ごま塩状の雑音となって現れることもある。与えられた図形を整形したりその形状の特徴を集約した点図形やを求めるためにこのような誤差はできる限り低減できることが望ましい。画像処理手法の中でこうした 2 値画像の整形時に発生する誤差への対処として主なものに膨張・収縮処理が挙げられる。膨張 (Expansion) とは 1 画素分大きくする処理であり、拡張 (Dilation) や伝搬 (Propagation) と呼ばれている。一方、収縮 (Contraction) とは与えられた連結成分の境界画素をすべて取り除いて逆に 1 画素分小さくする処理であり、侵食 (Erosion) とも呼ばれている。Fig.12 に膨張・収縮処理の概念図を示す。

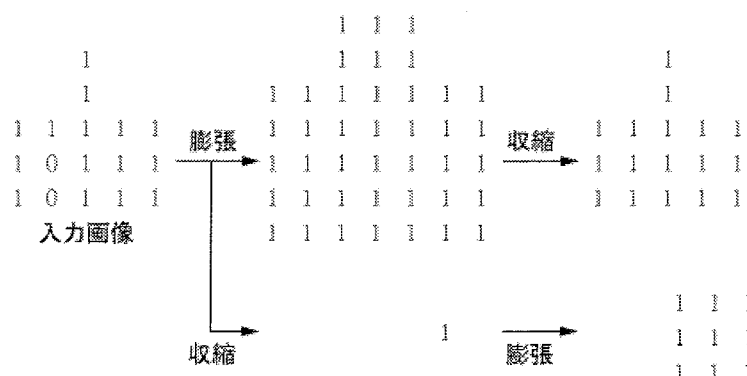


Fig.12 膨張・収縮処理の概念図

入力画像を見ると、凸型の塊に 2 画素分に値 0 の画素が存在する。処理の方針としてすべて 1 である方が望ましいようなとき、膨張・収縮処理が適している。上の処理では膨張によって画素の周りが 1 画素ずつ大きくなり、収縮によって元のサイズに戻すと入力画像で存在した値 0 の画素が消滅していることが分かる。しかし、下のように処理を逆にした場合は全く別の形状になってしまうため、データの性質を見極めて処理を行う必要がある。(ごま塩状の雑音の場合は収縮・膨張処理の方が適している場合が多い)。膨張・収縮処理は以下の条件に沿って処理を行う。

- 膨張 任意の画素 (i,j) あるいはその4近傍のいずれかが値1のとき (i,j) は値1、その他のとき (i,j) は値0
- 収縮 任意の画素 (i,j) あるいはその4近傍のいずれかが値0のとき (i,j) は値0、その他のとき (i,j) は値1

Fig.13 に実際のレーダ観測データに動的背景差分、2 値化処理、膨張・収縮処理をした処理結果を示す。



Fig.13 動的な背景差分処理、2 値化処理、膨張・収縮処理を行ったレーダ観測データ

2.4.3. ラベリング

前述した2値画像において、画像中の図形を分類するために同じ連結成分に属するすべての画素に同じラベル(番号)を割り当て、異なった連結成分には異なったラベル(番号)を割り当てる操作を、連結成分のラベリングという。ラベリング操作は、個々の連結成分の属性の解析に先立って各成分を抽出するために不可欠の操作である。Fig.14に2値画像にラベリング操作を行う概念図を示す。

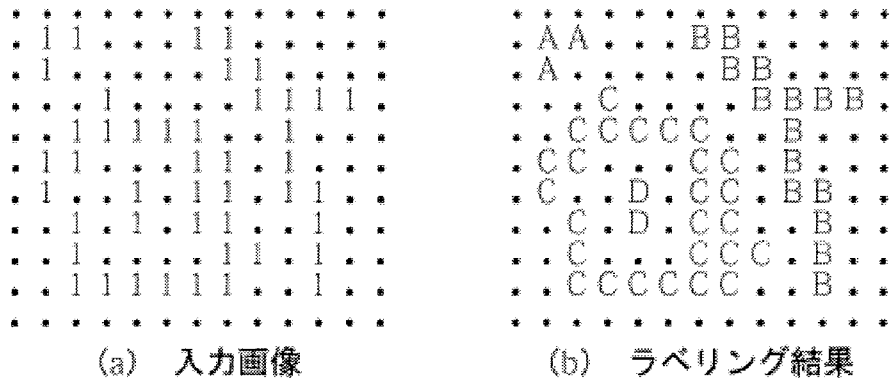


Fig.14 ラベリングの例

Fig.14において、(a)の格子状に並んだ点・数字は画素を示す。ここでは便宜的に点の画素は0を表しており、1は2値画像のもう1つの値を示す。1の値を持つ画素のうち連結しているものを1つの塊とする場合、ラベリング処理は以下の手順で対応付けを行う。

手順1) スキャン(走査)はラスタスキャンと呼ばれる画像の上の行から下の行に向かい1行ずつ、左から右側に操作する順序で行われる。スキャンした画素の上隣の画素が0の値を持ち、すでにラベルが付けられていれば、現在の画素にもそれと同じラベルを付ける。また、その画素の左隣の画素も0の値を持ち、かつ上隣の画素とは異なるラベルが付けられていれば、同じ領域に異なるラベルが付けられたことになるので、その2つのラベルを記録する。

手順2) 上隣の画素が1の値を持つ場合、左隣が0の値を持ち、すでにラベルが付けられていれば左隣と同じラベルを付ける。

手順3) 上記のいずれの画素も1の値を持つとき、新しいラベルを作り、現在スキャンしている画素にそのラベルを付ける。

手順4) 手順2によって得られた重複関係を基に、ラベルを一番小さい値に合わせる。

以上の処理により得られた画像が (b) となる。値 1 を持つ画素同士が A, B, C, D の四つのグループに分けられていることが分かる。またここではアルファベットを用いてグループを表現しているが、実際には数値で区別されることが多い。

2.4.4. 船舶候補領域の対応付け

前述した手法によりレーダ信号の反射が強い領域についてラベル付けを行ったが、ラベリングにより得られた画素の塊のうち、動的な背景差分法で除去しきれなかった Sea Clutter や Weather Clutter、空中浮遊物かの反射が残ることがある。

ある程度の安定的な反射面積を持ち、隣接画素も一定の輝度を有する船舶の反射に対し、このような Clutter は位置が定まらず、反射の面積も極端に小さいことがわかっていいため、本研究では有意な面積を持つものを船舶候補領域としている。移動する船舶候補領域の追跡、すなわちフレーム間の同一船舶の対応付けには最近傍法を用いている。最近傍法の探索は処理効率を上げるために画像全体で探索を行わず、船舶の動きに即して次フレームでの位置を予測し、その画素の周りのエリアから探索する。

5 分間（30 フレーム）のレーダ画像から得た領域に、追跡結果を重ねた図を Fig.15 に示す。また、アルゴリズムにより得られた航跡データを緯度経度と同じ直交座標系式に変換した図を Fig.16 に示す。

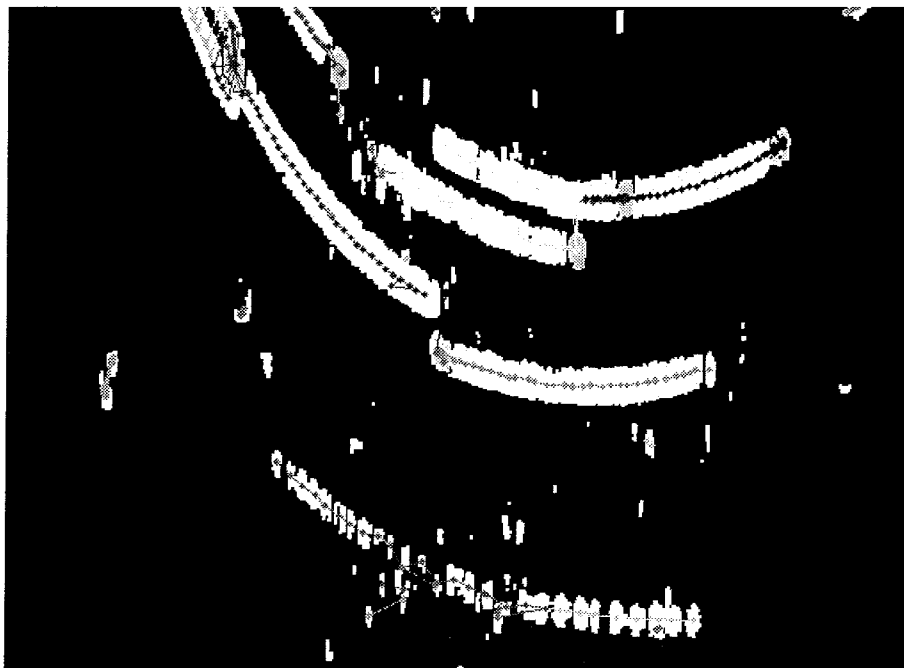


Fig.15 最近傍法による船舶のフレーム間対応づけ

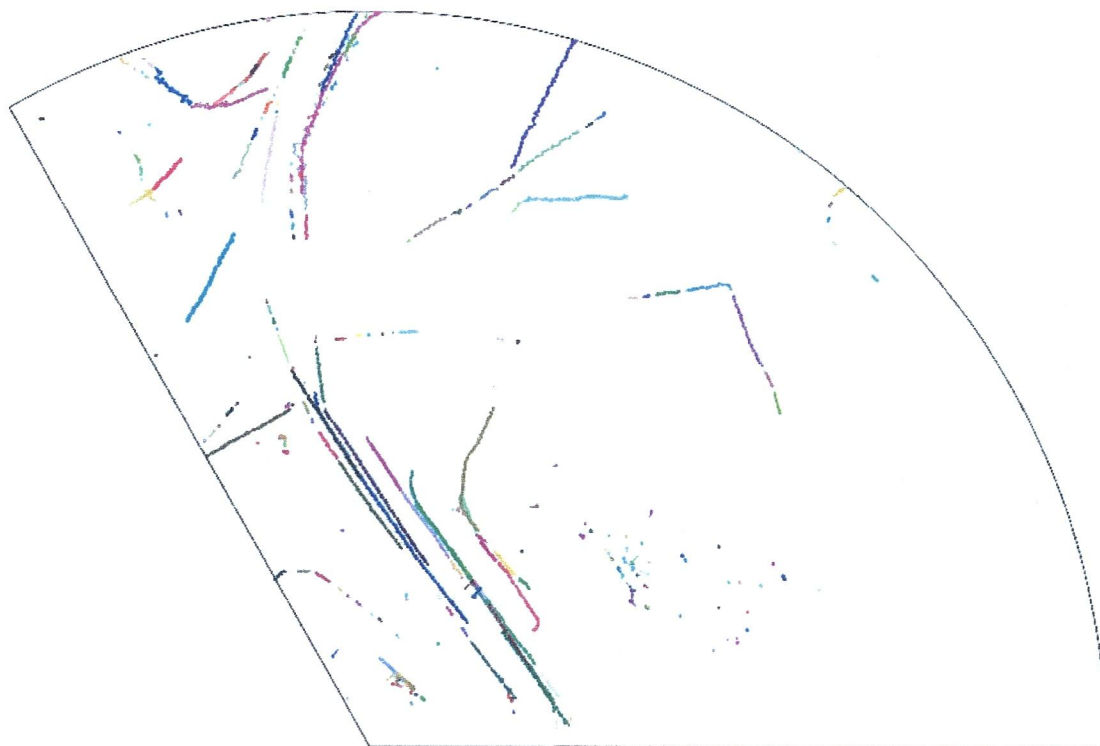


Fig.16 航跡の抽出結果

3. データベースの実装

3.1. 開発環境

本研究でのデータベースを開発した OS、プラットフォームなどの環境は以下の通りである。

OS: Windows XP professional version2002

プラットフォーム: Microsoft Visual Studio .NET 2003 MFC (Microsoft Foundation Class)アプリケーション

3.2. GUI による実装

本研究では個々の航跡をユーザにとってできるだけ扱いやすくするため、上記の環境で GUI (Graphical User Interface)を用いてデータベースの実装を行った。Fig. 16 にデータベースの基本操作画面を示す。Fig.17 の通り、実装したデータベースの構成は左に操作画面として「検索」グループボックス、「航跡」グループボックス、下部に「表示画像情報」グループボックス、右下に「ツリー構築」グループボックスに分かれており、右上はレーダの観測データの表示で構成されている。基本的な機能は主に海域・時間別の航跡の検索・表示と次章で述べる検用インデックスの作成で、グループボックスごとに操作の内容が分割されている。

使用法の概要はまず初めにユーザは「検索」グループボックスから検索に用いるインデックスを選択し、ファイルを開く。レーダ観測データ上にマウスで矩形を描き検索すると、データベースはインデックスファイルの中から該当する航跡データの ID を抽出し、格納して表示する。緯度経度と検索時間の範囲指定は同ボックスのエディットボックスやスクロールバーからでも可能である。検索結果は右側のレーダ観測データ内にその航跡を描画し、「航跡グループボックス」該当した船舶の数、各船舶の ID 番号が表示される。「表示画像情報」グループには現在表示されているレーダ観測データの取得した日時（年月日および時刻）、レーダ局、取得方位（レーダ局から見た方角）、レンジ、画像サイズが表示される。

船舶を描画する画像が海図ではなくレーダ画像である理由は、2.2 に記述した航跡抽出における認識（パターン認識）の問題にある。ここで描画される航跡は自動抽出により得られたものであるから、航跡抽出を目視で行った場合の航跡と比較して同様に抽出されたものと、途切れや誤検出を含んだものが存在する。これらを目視での抽出結果と同様に扱うことは、前述した通り海上交通解析に必要なデータを得る場合に障害となる場合とならない場合がある。データベースではこれらをレーダ画像と重畳表示することにより、ユーザが必要と判断すれば航跡抽出の正否をその場で確認することができる仕様を取っている。

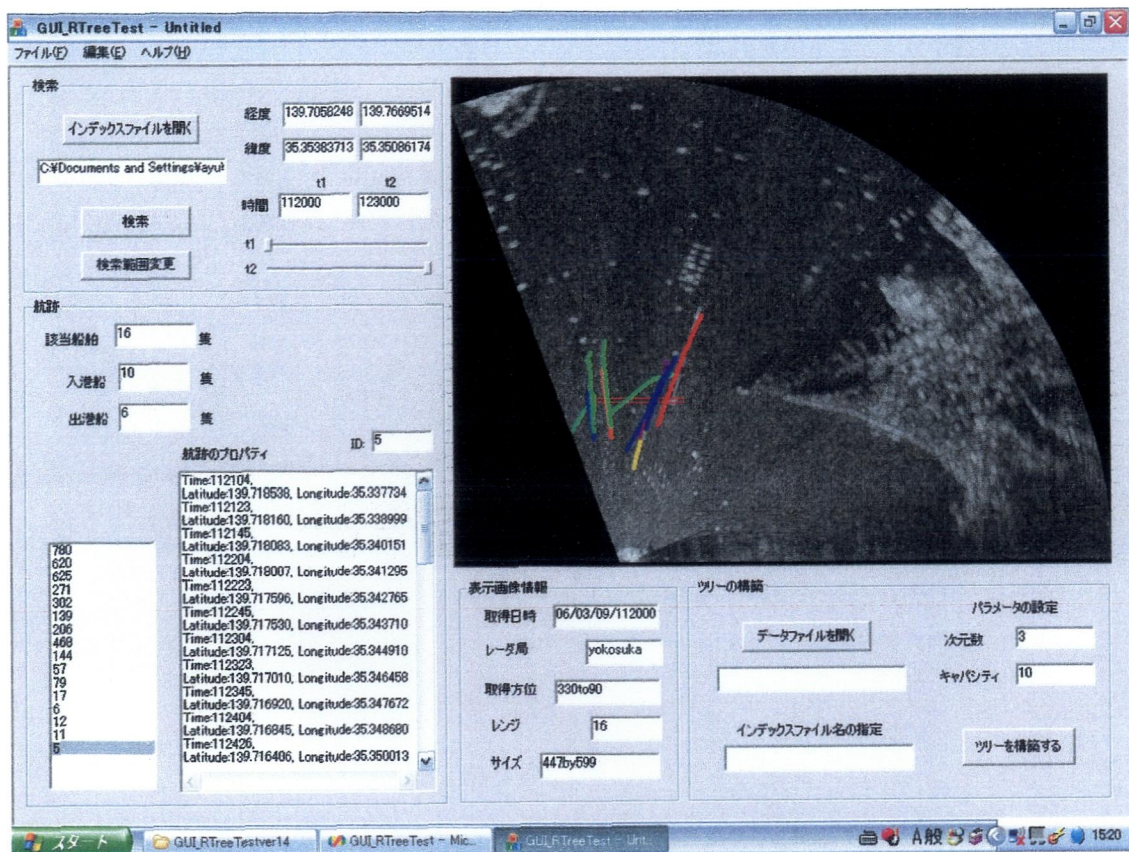


Fig.17 データベースの基本操作画面

3.3. 海上交通観測のための航跡検索

本データベースは海上交通観測に役立てるためのツールとして以下の値もしくは航跡を検索する機能を持つ。

- ① 検索した航跡の隻数
- ② 検索した航跡の隻数のうち、入港船と出港船の隻数および割合
- ③ 指定した ID に対応する航跡の団体表示、および個別表示
- ④ 航跡の ID 番号およびそれに対応する航跡のプロパティ

海上交通解析における重要なデータのひとつとして、交通量の調査がある。時間帯別にある地点を通過する船舶の数を数える際には、本データベースの①の機能を使うことができる。前節で述べた検索用の矩形は海域の検索の検索に用いられていたが、この矩形を狭く描けばそのポイントを通過する船舶の数とその動向を把握することができる。Fig.18 にある海域を通過した船舶の描画結果、Fig.19 に船舶の隻数表示の様子を示す。

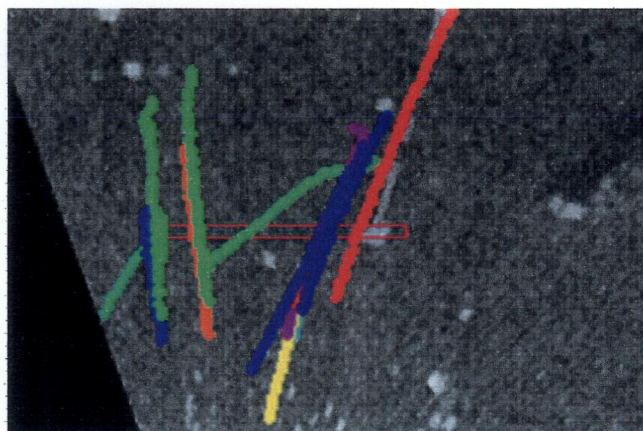


Fig.18 矩形内を通過する船舶の検索結果

航跡		
該当船舶	16	隻
入港船	10	隻
出港船	6	隻

Fig.19 通過した船舶の隻数表示

Fig.18 において、赤色の細長い矩形がユーザによって指定された海域である。航跡の検索はこの矩形に重なりがみられるものを該当とみなしているため、細長い海域（つまりある地点）を通過した船舶を②の機能からカウントできる。

さらに検索に該当した航跡について、各航跡を個別に表示することも可能である。Fig.20 のようにエディットボックスに表示された検索に該当した船舶の各航跡の ID 番号を指定すると、③の機能としてその船舶が個別に描画され、またその番号航跡の辿った緯度経度がプロパティとして示される。

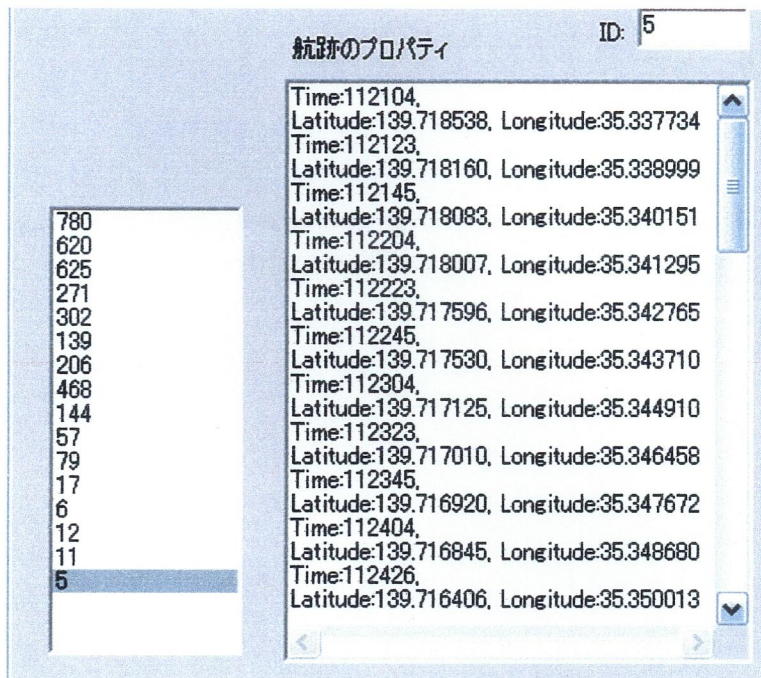


Fig. 20 航跡のプロパティ表示

4. 木構造を用いた航跡データの管理

4.1. 木構造の航跡データへの適用

航跡データの数には莫大な数にのぼる。前述したデータベースでは、指定した航跡を検索しているが、その操作性を重視すると検索は高速でなければならない。そこで本研究では航跡の管理に木構造を用い、検索の対象とする航跡に対してあらかじめインデックスを作成した。木構造とはグラフ理論の木の構造をしたデータ構造のことであり、階層構造とも呼ばれる。

一般に、地理情報システムの分野では地図情報などの空間データを扱う際に様々な木構造が用いられている⁽⁹⁾⁽¹⁰⁾。現在においては大縮尺数値地図データの整備に伴い、家形や道路沿などの詳細な地物を格納するために、大容量の情報を空間的に分割し木構造で管理する手法が多数提案されている。一方、航跡データは Fig.21 に示すように緯度・経度で構成される平面に時間軸が加わった空間データであるから、各航跡を最小単位とすると、同様に空間的な包含関係があるとみることができる。本研究ではこの点に着目し、高速な検索を実現するために木構造を用いた航跡の管理を行った。

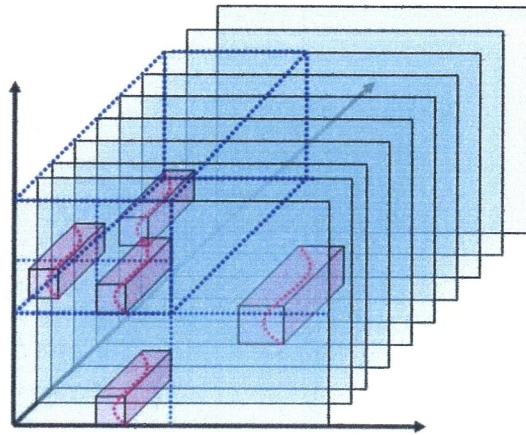


Fig.21 航跡データのデータ構造イメージ図

この研究では航跡データの管理に R*-Tree⁽¹¹⁾を用いている。R*-Tree は Beckmann らが提唱し、R-Tree⁽¹²⁾をベースに改良された木で、検索をより効率的にするためにいくつかのパラメータを最適化し、新たな Insert（データの追加）方法を提案した木の構築手法である。木構造は様々な種類が存在するが、本研究は航路上に集中的に存在し、海域によって数に差が出るという航跡データの性質から、データの密度によって重みを付けることができ、また検索時にデータアクセス量を少なくするよう木を構築できる R*-Tree を用いた。

4.2. 代表的な木構造

木構造はグラフの種類の一つで、単連結で閉路を持たない無向グラフのことである。根 (root) を起点として節点 (node) に枝分かれ (edge) している。枝分かれの元になっている節点を親ノード、枝分かれの先を子ノード、同じ深さの階層のノード同士を兄弟ノードと呼ぶ。子ノードの内、末端にあるものを葉ノードという。無向グラフとは節点から節点を結ぶ枝に向きの決まりが無く、親ノードから子ノード、もしくはその逆も通れるグラフのことを言う。Fig.22 に木構造の概要図を示す。

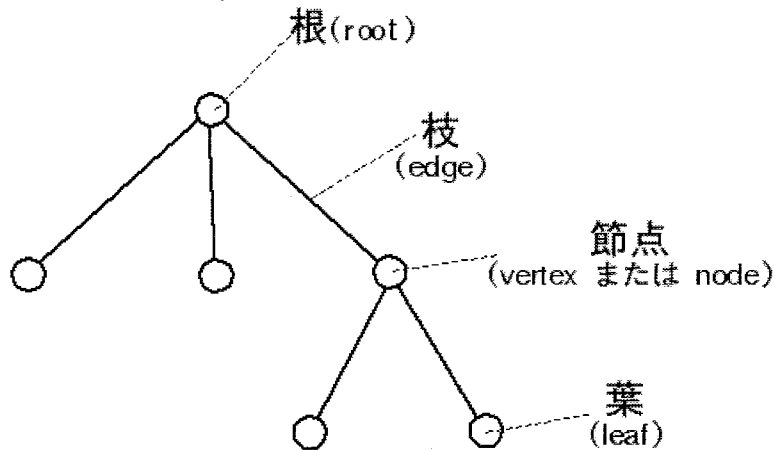


Fig.22 木構造の概念図

グラフ理論は数学の一分野として発達したもので、木の成り立ちや枝分かれの性質によって様々な木が考案された。後に木構造はコンピュータのデータ構造として用いられるようになり、コンピュータが扱うデータの記述方式や、インターフェースで使用されているファイル管理のためのディレクトリやフォルダも階層構造である木構造の概念に基づいて作成されたものである。木構造の種は多様でその特性に応じて分類され、主なものに二分木、多分木、順序木、二分探索木、平衡木などがあり、あるいは複数の特性を兼ね備えた木構造も存在する。二分木は一つのノードが二つの枝を持つ木、多分木は三つ以上の枝を持つ木である。順序木は一つの親ノードが持つ複数の子ノードに順序関係がついている場合の木で、二分探索木は親と子の間の値に大小の制限を持った木である。平衡木はすべての葉ノードの深さが等しい木のことを指す。航跡データの分割は順序や値に制限は無いため、単純な階層構造である木が適している。また木の大きさや形状は検索時間に大きく影響するため、できるだけ浅い平衡木が望ましい。これは木の階層構造が深いと検索に時間を要し、木の深さが均等でないと検索時間にばらつきが出てしまうためである。

本章では平衡木の中でも代表的な B-Tree⁽¹³⁾と R*-Tree の基となった R-Tree について述べる。

4.2.1. B-Tree

B-Tree は、Balanced Tree (バランス木) の略で、Binary Trees (二分木) を改善した手法として 1970 年代に登場して以来、長年にわたってデータベースやファイルシステムなどで利用されている。データの分岐 (枝分かれした「葉」) の先がすべて同一の階層に属しているような構造モデルのことである。B 木では、葉を除いた全ての節が一定数以下の分岐 (枝) をもち、その先が分岐した元の枝よりも少ない分岐を持っている。

B-Tree のさらに基となる Binary Tree (二分木) は、木構造アルゴリズムの中で最もシンプルな構造である。各ノードは 0~2 つの子ノードを持つ。オリジナルの二分木から改善された点としては、親ノードが複数の子ノードを持つ点、兄弟ノードの値が順に並んでいるであるという点にある。そのため検索速度が速く、実行時間が入力の大きさ n に比例するアルゴリズム $O(n)$ の場合、 $\log(O(n))$ で行うことができる。

時空間検索への拡張が難しい他、オーバーフローが無く、インデックスのパラメータによっては兄弟ノードの数が多くなり木構造の意味をなさないことになり、航跡データのインデックスに対しては不向きである。これについては 5.4 で詳しく後述する。

4.2.2. R-Tree

R-Tree は 1984 年に提唱された手法で B-Tree にオーバーフローを持たせ、 n 次元に拡張した平衡木である。二次元以上の空間領域について最小外接矩形 (MBR: Minimum Bounding Rectangle) を定義している。親ノードは葉ノードを除きは少なくとも 2 つ以上、 m 個以上 M 個以下の子ノードを持ち、MBR を使って各点 (または図形) を仕切るが、MBR 同士はオーバーラップがあってもよいというルールに従って構築される。

R-Tree のインデックスでは木を探索することで目的の点を探すので、探索のコストは平均で $O(\log n)$ 程度であるが、最悪時には $O(n)$ である可能性があるため、さらに高速化する条件が必要である。一方、インデックスに必要なメモリは、 n 個の点に対する木を作るので $O(n \log n)$ 程度である。R-Tree、もしくはそれから派生した手法のインデックスは MBR の作り方の柔軟性が高く、実用化されているデータベースでの空間インデックスとして広く使われている。例として空間上に散らばる点に R-Tree を適用した場合の MBR の定義と、木構造に格納した場合の図を Fig.23 に示す。

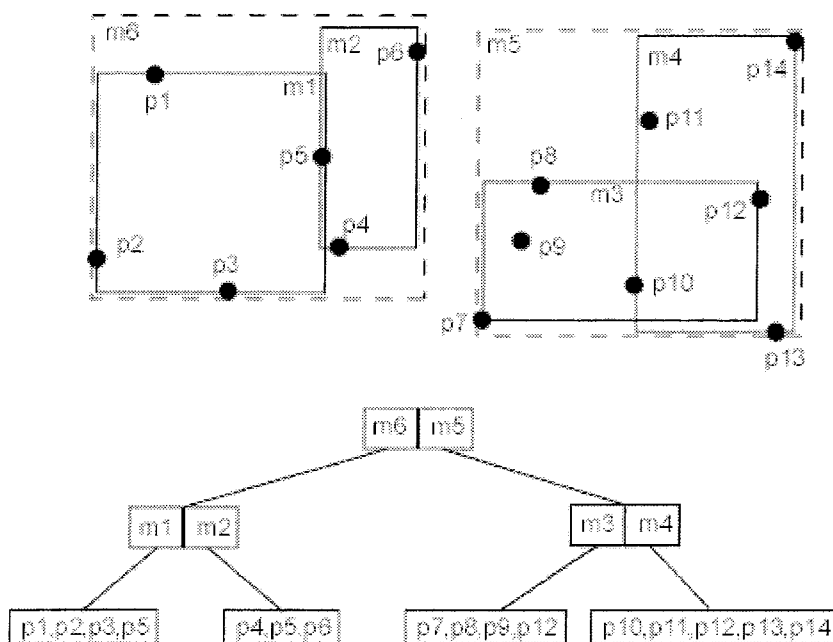


Fig.23 R-Tree と MBR

4.3. 領域分割と木構造

R-Tree を基本とした空間データ管理への応用の基本原理は、空間内を分割し、距離空間的距離の近い個々のデータを同じグループとして扱い、さらに分割された領域内で分割を繰り返すことで位置関係に包含関係を持たせ、階層的に管理するというものである。Fig. に R*-Tree (この点は R-Tree でも同じ手法) を航跡データに適用した場合の包含関係の概要図を、Fig.24 にそれらを木構造で表現した場合の概要図を示す。

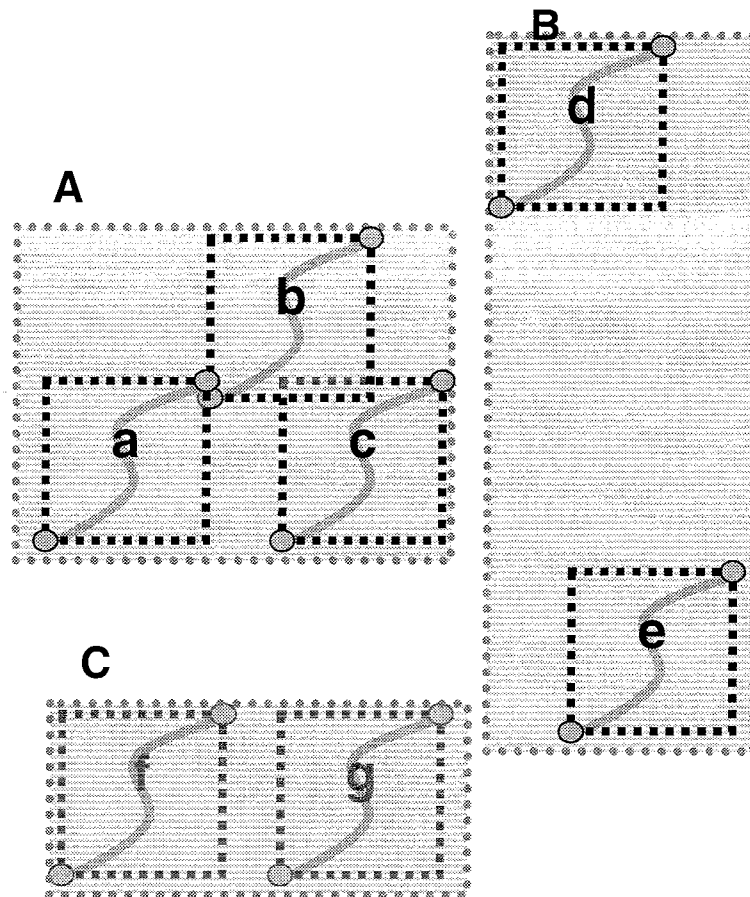


Fig.24 航跡データ分割の概要図

Fig.24 は a,b,c,d,e,f,g のような 7 つの航跡があった場合の例である。説明を簡略にするためにここでは二次元の図に置き換えている。緯度経度からなる座標の集合である各航跡は外接する矩形領域（最小外接矩形）を持ち、これは木構造で管理する領域のうち最小の単位とする。空間内では空間的に距離の近いもの同士を同じ領域にまとめて管理する。Fig. では全体が領域 A,B,C のように分割され、航跡 a,b,c は領域 A に含まれているような包含関係がある。木構造による空間データ管理ではこの包含情報を Fig. のように木構造の情報に格納する。

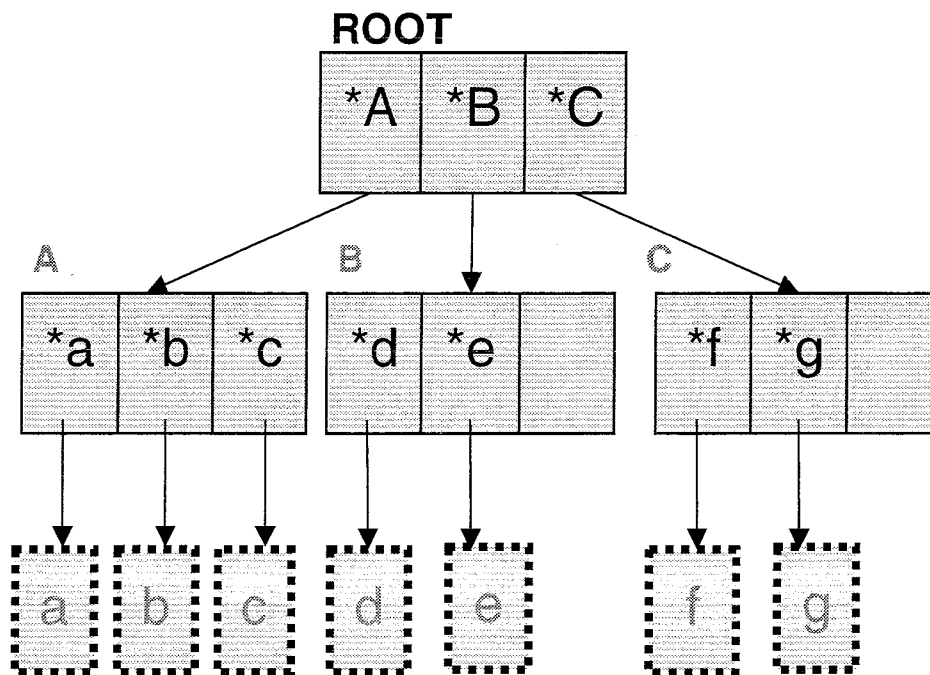


Fig.25 航跡データの木構造への格納

Fig.25 では、ROOT がデータ全体をさす。ROOT は領域 A, B, C に分割されているので、実線枠で示したスロット（データ格納先、プログラム内では構造体で定義した変数）へ子ノードである領域 A, B, C へのポインタが格納される（文字の前の*はその値へのポインタを差す）。さらに領域 A のスロットには、領域 a,b,c のポインタが書き込まれていることが分かる。子ノードに格納されているのはポインタのみであり、検索結果として表示するための航跡データ、つまり緯度経度や時間などの正味のデータは点線枠で示した葉ノードである a,b,c,...g に格納されている。

また、領域 B, C のスロットには空きが見られる。木構造の構築は最初に小さな木から始まり、航跡データを木構造を格納するルールに従い、データを木に投入しながら空きのスロットを埋める。データがスロットに収まりきらない場合などは新たに枝を伸ばし、木を大きくしていく。次節 5.4 ではこうした木の構築法と、R-Tree よりもさらに検索を高速化した R*-Tree の特性について述べる。

4.4. R*-Tree によるインデックスの作成

4.4.1. 木構築

グラフ理論ではノードにあたる、コンピュータのデータ構造ではスロットの塊であるデータを格納する媒体をストレージと呼ぶ。R*-Tree では一つのストレージ内のスロットの数を固定している。木の構築（各 MBR の領域分割）は、初めに空間内に散らばる MBR を決められたスロットの数に応じて領域を分割する。このとき分割は原則として MBR 同士の距離が近いものとされているが、以下の点についても考慮されている。

- MBR を含む矩形領域の面積を最小化する
- 矩形領域同士のオーバーラップ（重なり）の面積を最小化する

上記二点は R-Tree から引き継がれてきた性質であるが、これには検索効率を上げるための利点がある。まず MBR を含む矩形領域は広ければ広いほど検索する領域 (Rectangle enclosure query、以下レンジクエリ) に重なる可能性がある。たとえレンジクエリに航跡の MBR が存在しなかったとしても、MBR を分割している矩形領域に重なった場合は検索の際木を辿る（トラバース）ことになる。従って、MBR 以外のスペースはできる限り少ない方が望ましい。

また上記と同様の理由で矩形領域同士のオーバーラップも最小化する。求める MBR 一つに対し矩形領域に重なりが存在すると検索は二通りの枝をトラバースしなくてはならないためである。以上二点の概要図を Fig.26, Fig.27 に示す。

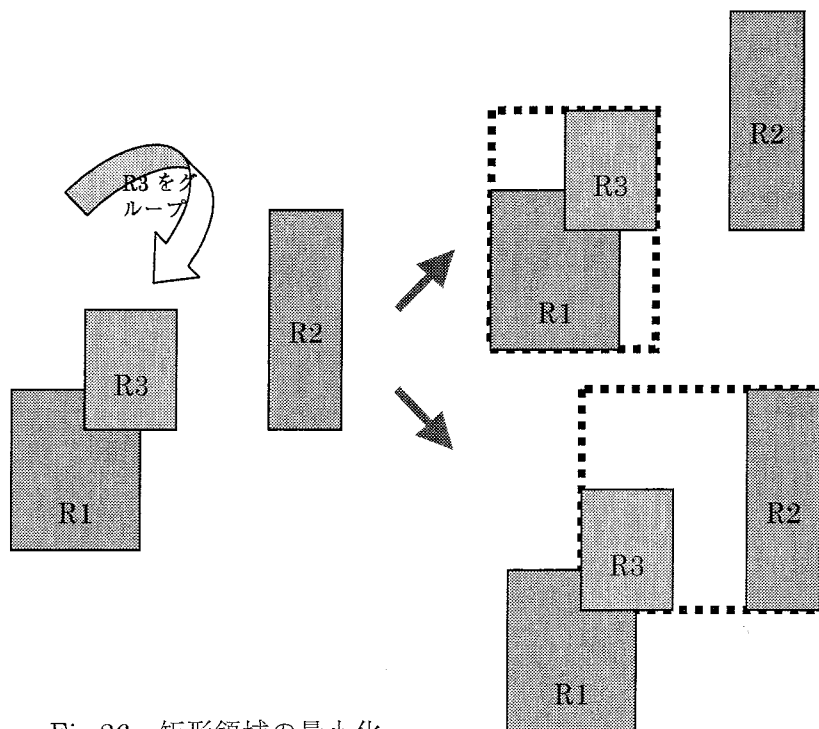


Fig.26 矩形領域の最小化

Fig.26 において、矩形領域 R1, R2, R3 は R2, R3 の領域よりも R1, R3 の領域の方が無駄が少ないことが分かる。

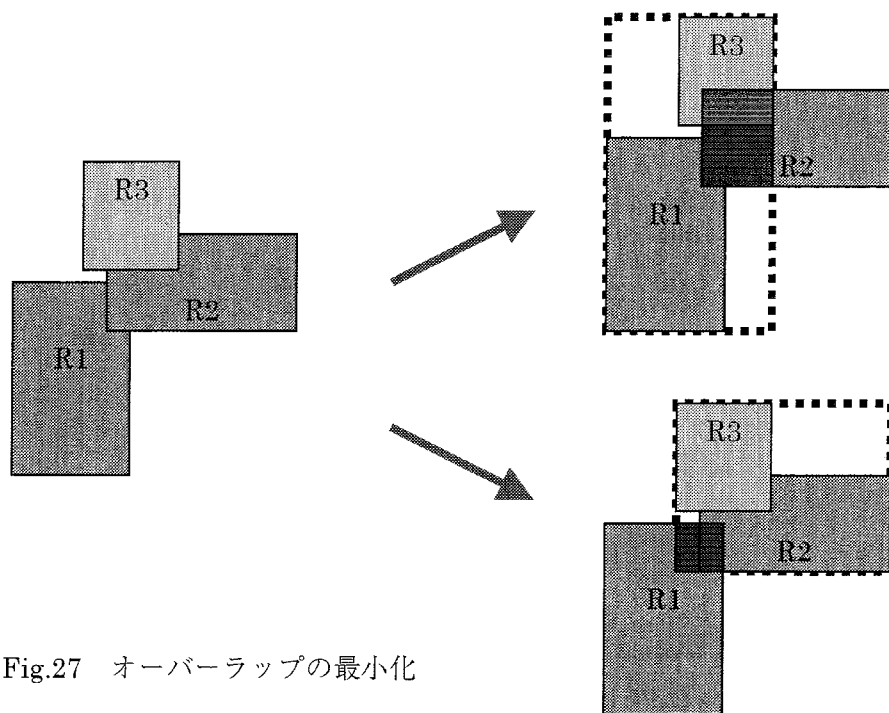


Fig.27 オーバーラップの最小化

Fig.27 においても、R1, R3 の領域よりも R2, R3 の領域の方が面積が小さく、括り方によってオーバーラップに差がでる。これらの矩形領域やオーバーラップの面積が少ないほどレンジクエリが重なる確立が下がり、無駄な検索を省くことができる。

ここまでは分割の方法について述べてきたが、木の構築には分割した領域の中でさらに分割する必要性が生じた場合（木構造ではスロットの中にデータが入りきらないオーバーフローの状態）も起こる。R*-Tree ではオーバーフローが起こった場合は以下のようにして再挿入、または新たな分割を行う。例として Fig.28 のような状態を挙げる。

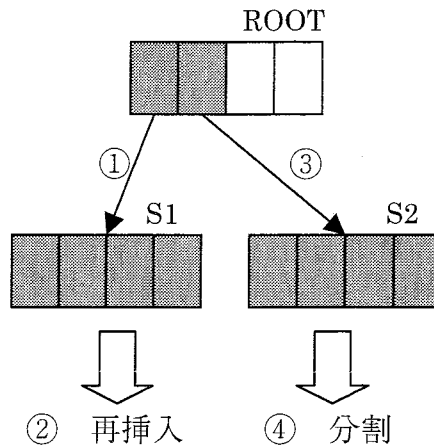


Fig.28 オーバーフロー

オーバーフローを判定するためには二段階の条件がある。①のようにまず ROOT に投入されたデータが前述した条件に従って R1 のストレージに入ってきたとする。S1 のスロットはすべて埋まっているので、データはここで②の再挿入を行う。再挿入とは S1 に格納された M 個のデータ（ここでは 4 つ）と新しく挿入されたデータ（1 つ）の中から S1 とは別のストレージへ移す手法のことである。再挿入では $M+1$ のデータ（エントリ）のすべてに対し、そのストレージの矩形領域の中心と、それぞれのエントリの中心距離を計算する。その後最も距離が M 個のエントリまでを S1 に格納し、取り除かれた 1 つのデータを新規データとし、③のように初めから木に投入する。再挿入によってはじかれたデータが他のストレージに収まった場合はそれまでとし、さらにそのデータがオーバーフローしてしまった場合に④分割が行われる。

4.4.2. R*-Tree の特徴

R*-Tree は 5.1 で前述したように、R-Tree を検索を高速化するために改良されたものである。R*-Tree では木の構築を最適化する。探索時間のばらつきを無くすため、構築は平行木とし、木のストレージの利用率が高くなるように設定する。スロットの数は予め決められているため、ストレージの利用率が低ければ巨大な木になってしまい探索に時間がかかるためである。本論文ではスロットの数は 10 に設定し、時間と矩形領域の座標を記録したテキストファイルからインデックスファイルをバイナリデータ形式で作成し、データベースに入力して用いる。レーダ観測データに映る船舶は、航路内の航跡同士は重なりやすく、また一方で船舶が全く通らない海域も存在するため、範囲（海域）によって航跡の存在する密度に差が出てしまう。よって線形探索のように端から順に検索することは非常に効率が悪いといえる。木構造では航跡の集中する海域に領域分割を多く行い、少ない地点では領域の分割を最小限に抑えられるため交通量が少ない海域への過剰な探索時間が短縮されることが期待できる。

4.5. 実験 線形探索法との時間的効率における比較

本論文では木構造を用いたインデックス作成の有効性を検証するため、指定した航跡データを線形探索した場合とインデックスを用いた場合の探索時間を計測する実験を試みた。実験の実施要領は以下の通りである。

- (1) 各実験ともに任意の時間・緯度・経度の条件に含まれる航跡の検索を行い、検索に掛かった時間、該当した船舶の隻数を記録した。
- (2) 各航跡データとも試行回数は 20 回行った。
- (3) 木構造を用いた検索に関しては、インデックスの作成に要した時間も計測した。

線形探索による実験では、Fig.29 で示す航跡データの先頭から、(1) の条件に該当する船舶を順に調べ、条件に適った船舶の ID 番号を保存している。このデータは Table1 に示すような多数の属性を持つ航跡データを検索用に加工したもので、船舶の ID 番号と MBR の情報のみから構成されている。この MBR データはテキストファイル形式で、7つの属性からなる。左から順に 1~2 桁の番号が航跡の ID 番号、MBR の始点 (Top Left) の経度、MBR の終点 (Bottom Right) の経度、MBR の始点の緯度、終点の緯度、始点が観測されたレーダ観測データ上の時間 (時、分、秒)、終点が観測された時間となっている。

木構造によるインデックス作成はこのような空間的データの最小の情報に要約し、4.4 で述べたような領域分割を行い各領域を管理している。従って実際の木構造内における検索の対象は最小外接矩形であり、データベースに表示させるための航跡データや時間ごとの緯度経度情報などは検索の終了後に条件に該当した船舶についてのみ詳細情報にアクセスさせている。今回は両手法の公平性から線形探索にもこの最小外接矩形を用いた。

また木構造によるインデックスを用いた実験では、前述したデータベースを用いて線形探索による実験と同じ時空間領域を指定し、それに該当する MBR の検索時間を計測した。

radar1 - メモ帳									
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)									
0	139.8176	139.8268	35.4423	35.4513	112022	112504			
1	139.7103	139.7317	35.2998	35.3257	112022	112926			
2	139.7327	139.7424	35.3007	35.3341	112022	113126			
3	139.7468	139.7638	35.2634	35.2830	112022	112726			
4	139.7794	139.7865	35.4088	35.4156	112042	112323			
5	139.7131	139.7185	35.3377	35.3834	112104	113329			
6	139.7288	139.7564	35.3236	35.3686	112104	113829			
7	139.8789	139.8844	35.4824	35.4870	112123	112323			
8	139.7057	139.7195	35.3813	35.4418	112123	113608			
9	139.7052	139.7407	35.4035	35.4348	112123	113329			
10	139.6725	139.6845	35.3810	35.3891	112123	112907			
11	139.7025	139.7068	35.3348	35.3777	112123	113329			
12	139.7414	139.7623	35.3358	35.3744	112123	113829			
13	139.7889	139.7975	35.4551	35.4591	112204	112404			
14	139.8408	139.8434	35.3980	35.4032	112204	112426			
15	139.7286	139.7425	35.4280	35.4458	112204	112926			
16	139.6648	139.6759	35.3885	35.3967	112204	113007			
17	139.7067	139.7071	35.3461	35.3545	112204	112445			
18	139.7253	139.7431	35.2917	35.3116	112204	112907			
19	139.7282	139.7586	35.2683	35.3055	112204	113448			
20	139.7349	139.7603	35.2705	35.2988	112204	113408			
21	139.7528	139.7608	35.3618	35.3783	112245	112745			
22	139.6978	139.7020	35.3360	35.3426	112245	112504			
23	139.8667	139.8724	35.4282	35.4349	112304	112626			
24	139.6858	139.7154	35.3033	35.3512	112304	113829			
25	139.8099	139.8433	35.3601	35.3768	112323	113508			
26	139.7252	139.7292	35.3008	35.3053	112323	112645			
27	139.7030	139.7065	35.4072	35.4108	112345	112545			
28	139.8213	139.8418	35.3591	35.3616	112345	114251			

Fig29. 実験に使用した航跡データの例

今回の実験に使用した航跡データのもとであるレーダ観測データの取得日時、時刻、データの情報を Table1 に示す。ただし、属性項目は以下の状況を示す。

- レーダ局 取得したレーダ局。本実験では横須賀レーダ局。
- 西暦／月／日 データを取得した日時
- 開始時刻／終了時刻 データを取得した時刻。時：分：秒の形式で記載。
- レンジ 取得したデータのレンジ（レーダからの距離）
- フレーム間隔 センシングした間隔（秒）
- ゲイン レーダ信号のゲイン
- 気象条件 データ取得を開始した時点の天候。長時間にわたり特徴的な天候であった場合記録しており、空欄は晴天、もしくはうす曇りなどで時折日が陰る天候であった。

Table1

レーダ局	西暦	月	日	開始時刻	終了時刻	レン ジ	フレーム 間隔	ゲ イ ン	気象条件
横須賀	2004	9	6	15:07:58	15:20:23	6	20	10	
横須賀	2004	9	6	15:53:52	16:03:36	6	20	10	
横須賀	2004	9	6	16:36:40	16:46:24	6	20	10	
横須賀	2005	1	18	14:19:29	14:29:15	6	20	80	
横須賀	2005	1	21	14:02:24	14:10:39	6	10	128	
横須賀	2005	5	11	13:18:39	14:25:05	6	20	100	曇りのち晴 れ
横須賀	2005	5	11	12:03:06	13:09:51	6	20	100	曇りのち晴 れ
横須賀	2005	5	13	19:29:08	20:23:08	6	20	100	
横須賀	2005	5	14	17:17:48	18:24:34	6	20	100	
横須賀	2005	5	18	14:56:45	16:03:31	6	20	100	強風
横須賀	2005	5	22	18:42:24	19:50:51	6	20	100	雨天
横須賀	2005	5	23	10:21:33	11:28:17	6	20	100	干潮
横須賀	2005	5	23	17:07:00	17:53:37	6	20	100	満潮
横須賀	2005	12	6	20:55:23	22:02:07	9	20	100	
横須賀	2005	12	8	17:36:01	18:42:45	9	20	100	
横須賀	2005	12	8	14:46:52	14:55:09	9	20	100	
横須賀	2005	12	13	14:27:45	15:34:30	9	20	100	
横須賀	2005	12	13	18:18:28	19:25:14	9	20	100	
横須賀	2006	3	7	11:15:51	12:22:38	9	20	100	
横須賀	2006	3	9	11:19:22	12:26:09	16	20	100	
横須賀	2006	5	14	14:24:47	14:47:51	9	10	100	どんより曇 り
横須賀	2006	5	14	16:08:48	16:23:56	9	20	100	曇り
横須賀	2006	5	14	15:20:32	16:06:07	9	20	100	曇り
横須賀	2006	5	14	14:24:47	14:58:18	9	10	100	どんより曇 り
横須賀	2006	5	14	15:20:31	16:27:16	9	10	100	曇り
横須賀	2006	5	14	19:04:22	19:37:54	9	10	100	曇り
横須賀	2006	5	18	14:44:44	15:18:16	9	10	100	強風
横須賀	2006	5	18	20:12:21	20:45:51	9	10	100	曇り

横須賀	2006	5	20	13:50:58	14:07:39	9	10	100	晴れ
横須賀	2006	5	20	17:26:20	17:43:02	9	10	100	雨強風
横須賀	2006	12	12	08:42:50	09:16:24	9	10	100	曇り
横須賀	2006	12	12	10:55:10	11:11:51	9	10	100	曇り
横須賀	2006	12	12	12:55:45	13:12:26	9	10	100	曇り
横須賀	2006	12	12	13:40:22	13:57:05	9	10	100	雨天

Table2 に木構造を用いて作成した各インデックスの作成時間を示す。Table2 から航跡数が多くなるに従い作成時間が長くなっていることが分かる。ただし、レーダ観測データを取得する環境により、現時点では 1 日あたり最も長い時間観測できたのは 2006 年 5 月 14 日のデータで、この航跡数が 2194 隻であったため、同一日のデータは 2194 隻までとなる。航跡を検索する時間を計測するという観点から今回の実験では複数日のデータを組み合わせて 3220 隻以降のインデックスを作成している。インデックス作成の時間単位や航跡数については後述するが、おおむね 1000 隻ごとに増加させた規模のインデックスを作成し、その作成時間と検索時間の推移を調べた。これまでに取得することのできたすべての航跡を合わせた 6882 隻のインデックスが最多である。

Table 2 航跡数の違いによるインデックス作成時間の遷移

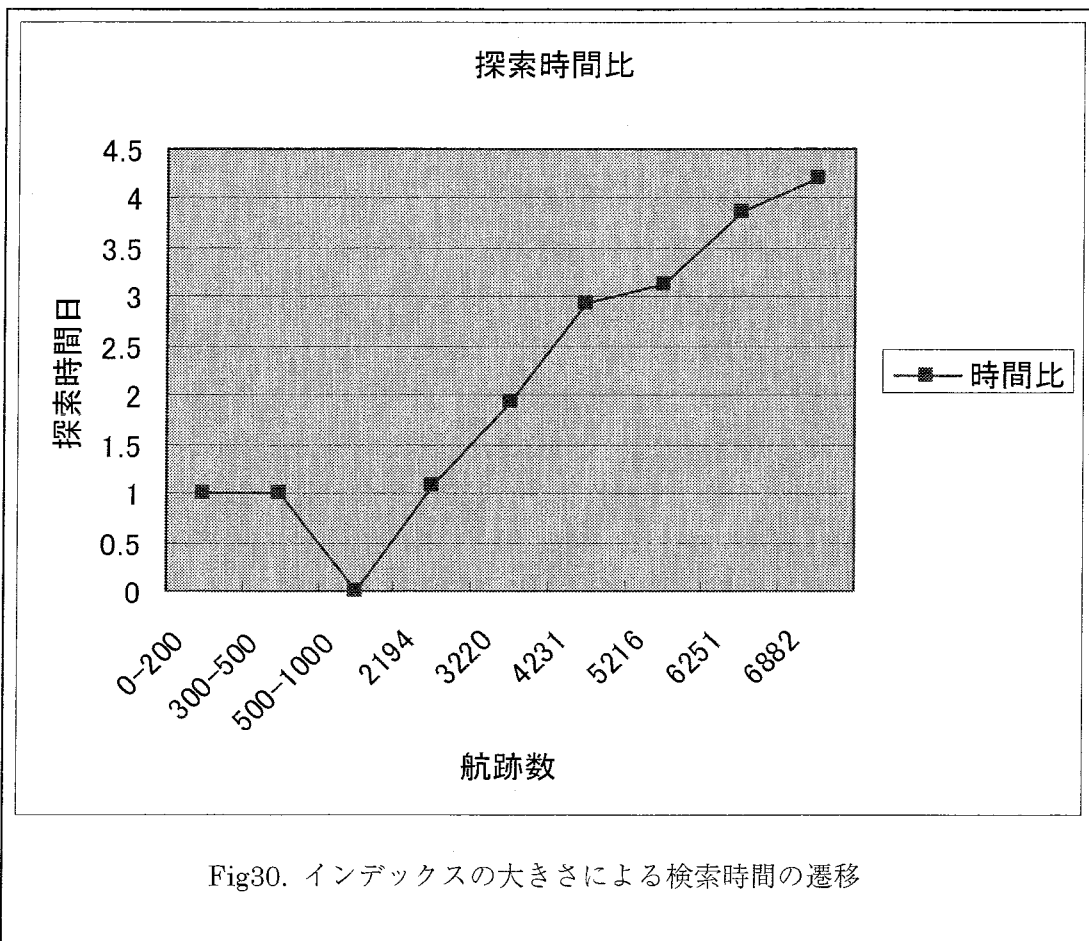
航跡数	インデックス作成時間 (秒)
32	0.063
34	0.046
55	0.109
96	0.188
116	0.203
125	0.234
173	0.296
179	0.343
309	0.547
356	0.625
378	0.61
422	0.704
532	0.875
604	0.71
832	0.563
2194	0.921
3220	1.359
4231	2.062
5216	2.437
6882	3.312

Fig.30 に線形探索と木構造を用いた検索の検索時間比を示す。グラフは各試行ごとに線形探索を木構造を用いた検索時間で割った値を計算し、試行 20 回の平均を取ったものである。航跡の検索は両手法ともに検索対象の大きさ（検索に該当する船舶の数）によって検索時間が異なるため、検索時間の平均ではなく検索時間の比を算出して比較を行った。また、各インデックスの航跡数に対する検索対象の大きさに対しても特性が見られる。例を挙げればインデックス内に存在するすべての船舶が該当するように検索を行えば木構造も線形探索も検索時間はほぼ同時間となり比較の必然性が見られないなどである。今回は実際にデータベースでユーザが検索した際に表示や解析データに混乱が見られない程度の航跡数を想定し、数隻から 100 隻程度までの船舶が該当するような検索範囲で検索を行った。

Fig.30 で見られるとおり、インデックスの大きさ（木の大きさ、航跡数）が多ければ多いほど、線形探索に比べ、木構造の方が早くなることが分かる。航跡数 6882 隻では 4 倍を越える速度での検索が可能であり、航跡の数が多ければ多いほど今後も差は増大していくものと考えられる。

しかし一方で木構造を用いた探索にはインデックスの作成に時間を要する。新たな航跡データが投入され、木が巨大になれば作成に時間も掛かってしまうため、本研究ではインデックスは実際の状況に即した大きさに区切るべきであると考えている。実験で使ったような日ごとのデータから、週、月、年ごと、など様々な単位で括ることができるが、これはデータベースで扱う対象の海域の交通量や、解析する際の事情に応じて設計されるべきである。

例を挙げれば、現在の航跡と 10 年前の航跡では港湾の細かな状況や経済・流通の種類も異なる可能性があり、海上交通解析の観点から見れば両者のデータが同一インデックスに存在する実質的な価値はそれほど高くないといえる。インデックスの具体的な大きさには今後議論の余地が存在するが、実際に交通解析を行う海事の専門家の意見を参考に、インデックスの作成条件を決定する必要がある。



5. おわりに

本研究では海上における安全策の一環として、自動的に抽出した航跡データを用い、海上交通解析を念頭に置いたツールとしてのデータベースを提案した。航跡追尾の自動化はパターン認識技術の特性においてレーダ画像処理の過程で生じる検出漏れ、誤検出、とぎれなどの現象のため、十分に正確な航跡を完全に得ることは難しい。しかしそのようなレーダ画像からの航跡であってもデータベースに蓄積し、データベース上から必要とする航跡の候補を示す機能をつけることで海上交通特性の一部である航跡の動向や数を確認することができた。

今後はデータベースの仕様の更なる充実に加え、3.2 で述べたようなレーダから離れるほど情報量が薄れるというレーダ観測データの性質に着目し、扇島レーダ局など他の局からの情報と連動させて、2局の観測範囲が重なる場所を補正し合うなど画像を重ね合わせたり、情報を補間しあうシステムの開発に努める。

参考文献

- (1) 新保雅俊, 平澤雅人, 大島正毅: 航海環境時系列画像からの航行船舶の抽出と追跡, 日本航海学会論文集, vol.113, pp115-126, 2005.9
- (2) 萩原秀樹, 大津皓平, 庄司るり, 田丸人意, 高橋宏直, 中場晃: AIS と予定航路情報に基づく新交通管理システム, 日本航海学会誌 NAVIGATION, 特集 pp21-25, 2004.3
- (3) 庄司るり, 萩原秀樹, 大津皓平, 林尚吾, 浪江宏宗: 東京湾におけるレーダネットワークシステムの研究, 日本航海学会論文集, Vol.109, pp.69-78, 2003.5
- (4) 庄司るり, 萩原秀樹, 大津皓平, 林尚吾, 浪江宏宗: 東京湾におけるレーダネットワークシステムの研究, 東京海洋大学学術講演会第一回論文集, pp99-102, 2004.2
- (5) 岡野匡, 萩原秀樹, 田丸人意, 庄司るり: 船舶交通解析におけるトラッキングの自動化に関する研究, 日本航海学会論文集, Vol.113, pp. 77-84, 2005.9
- (6) 中嶋鮎子, 庄司るり, 大島正毅: レーダ画像を用いたフロー解析による船舶の自動追跡, 日本航海学会論文集, Vol.113, pp. 43-51, 2005.9
- (7) 中嶋鮎子, 渡辺 信, 全 炳東, 大島正毅: 海事データベース構築の基礎検討, 日本航海学会第 114 回講演会予稿集, I-12, 2006.5
- (8) R. Pless, J. Larson, S. Siebers and B. Westover: "Evaluation of Local Models of Dynamic Backgrounds", Conference on Computer Vision and Pattern Recognition (CVPR '03), Vol. II, pp. 73-78, 2003
- (9) 根岸幸生, 大沢 裕: GBD 木のための空間インデックス高速初期構築法の提案, 日本データベース学会 Letters Vol.4, no.2, pp.5-8, 2005.9
- (10) 大沢 裕, 坂内正夫: 2 種類の補助情報により検索と管理性能の向上を図った多次元データ構造の提案, 信学論, Vol.J74-D-I, No.8, pp.467-475, 1991
- (11) N. Beckmann, H. P. Kriegel, R. Schneider and B. Seeger: "The R*-tree: an efficient and robust access method for points and rectangles", ACM SIGMOD, pp. 322-331, 1990.5
- (12) A. Guttman: "R-trees: A Dynamic Index Structure for Spatial Searching", Proc. ACM SIG-MOD, pp. 47-57, 1984.6
- (13) BAYER, R. "Symmetric binary B-trees. data structure and maintenance algorithms," Acta Inf. 1, 4 (1972), 290-306

謝辞

本研究を進めるにあたり、本当にたくさんの方のご協力をいただきました。卒業研究より始まったこのレーダ観測データを用いた研究は、未熟である私にとっては困難を極めたものであり、下記の方々のご指導、ご教授なしでは進めることができませんでした。

辛抱強く指導してくださった指導教官の大島正毅教授をはじめ、この研究では多くの先生方、学生の皆様にお世話になりました。海事分野の観点から、その楽しさと可能性を示してくださった庄司るり講師、お忙しい中時間を割いていただき本当にありがとうございました。航跡追尾の専門家である萩原秀樹教授には考えさせられるアドバイスをいただき、現実的な難しさを教えていただきました。レーダ研究メンバの一員として共に頑張ってくれた学部4年の山下健夫くん、大量なデータの処理は大変でしたがありがとうございました。修士1年の周立輝くんには技術的なサポートや学生同士のざっくばらんなディスカッション、研究のスパイスにさせていただいたと思っています。感謝しています。

そしてこの研究が千葉大学との共同研究になったことで、着手できる範囲が大いに広がり、とても有意義な時間を過ごすことができました。研究のアイデアをすばやく形にしてグループのギアを上げてくださった小林弘樹さんに感謝の意を表します。修士2年の立場で違う角度から同じ問題に取り組んだ渡邊信さんのいつも冷静に研究に取り組む姿勢と頑張りが励みになりました。最大の難関であった木構造の分野では頭脳明晰な宇都宮将陽さんにただいなお協力をいただきました。学部4年の池上雅昭さんはいつも穏やかですがこの研究に対して真摯に取り組んでおられ、今後も修士に進まれることをとても嬉しく思います。最後に、寝る間を惜しんで文章指導ならびに白熱したディスカッションをしてくださり、研究の方向性について大いに勉強させていただいた全炳東教授に本当に厚く御礼申し上げます。今後も修士で学んだ貴重な経験をもとに、何事にも取り組んで生きたいと思っています。

ここには書ききれませんが、3年半の研究生活を支えてくださった情報システム研究室の皆様、同期の院生の方々、ならびに大学の職員の皆様にも心より感謝申し上げます。ささやかではありますが、御礼の言葉と代えさせていただきます。